

# Linearly-Coupled Fountain Codes

Shenghao Yang, Soung Chang Liew, Lizhao You and Yi Chen

## Abstract

Network-coded multiple access (NCMA) is a communication scheme for wireless multiple-access networks where physical-layer network coding (PNC) is employed. In NCMA, a user encodes and spreads its message into multiple packets. Time is slotted and multiple users transmit packets (one packet each) simultaneously in each timeslot. A sink node aims to decode the messages of all the users from the sequence of receptions over successive timeslots. For each timeslot, the NCMA receiver recovers multiple linear combinations of the packets transmitted in that timeslot, forming a system of linear equations. Different systems of linear equations are recovered in different timeslots. A message decoder then recovers the original messages of all the users by jointly solving multiple systems of linear equations obtained over different timeslots. We propose a low-complexity digital fountain approach for this coding problem, where each source node encodes its message into a sequence of packets using a fountain code. The aforementioned systems of linear equations recovered by the NCMA receiver effectively couple these fountain codes together. We refer to the coupling of the fountain codes as a linearly-coupled (LC) fountain code. The ordinary belief propagation (BP) decoding algorithm for conventional fountain codes is not optimal for LC fountain codes. We propose a batched BP decoding algorithm and analyze the convergence of the algorithm for general LC fountain codes. We demonstrate how to optimize the degree distributions and show by numerical results that the achievable rate region is nearly optimal. Our approach significantly reduces the decoding complexity compared with the previous NCMA schemes based on Reed-Solomon codes and random linear codes, and hence has the potential to increase throughput and decrease delay in computation-limited NCMA systems.

## I. INTRODUCTION

Consider a wireless multiple-access network where  $L$  source nodes (users) deliver information to a sink node through a common wireless channel. Each source node encodes its message into multiple packets and transmits these packets sequentially over successive timeslots. All the transmissions start at the beginning of a timeslot, and the timeslots are long enough to complete the transmission of a packet.

Multiple access in such scenarios, where the goal of the sink node is to decode the messages of all source nodes, can benefit from *physical-layer network coding (PNC)* [1] (also known as *compute-and-forward* [2]) by decoding linear combinations of the packets simultaneously transmitted in each timeslot. Such a multiple-access scheme is called *network-coded multiple access (NCMA)* and has been studied in [3]–[5], where both PNC and multiuser

This paper will be presented in part at 2014 IEEE Information Theory Workshop.

S. Yang is with Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. Email: shyang@tsinghua.edu.cn

S. C. Liew and L. You are with Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China. Email: soung@ie.cuhk.edu.hk, yl013@ie.cuhk.edu.hk

Y. Chen is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China: Email: chelseachenyi@gmail.com

decoders are employed at the physical layer to obtain the aforementioned linear combinations. Specifically, Lu, You and Liew [3] demonstrated by a prototype that a PNC decoder can successfully recover linear combinations of the packets while the traditional multiuser decoder [6] that does not make use of PNC fails.

The ultimate goal of a multiple-access network is to recover the original messages of all users, rather than just the linear combinations of the transmitted packets among different users. Message decoding is hence required by NCMA to recover the original messages of all users. In this paper, we study this message coding problem induced by NCMA, illustrated as follows by a two-user multiple-access network.

#### A. Network-Coded Multiple Access with Two Users

Consider a wireless multiple-access network with two source nodes A and B. Nodes A and B transmit packets  $v_A$  and  $v_B$  simultaneously, and the sink node receives a superposition of the waveforms transmitted by both users. In the NCMA scheme in [3], two types of physical-layer decoders are used to decode the received waveform: 1) a conventional multiuser decoder that attempts to decode both  $v_A$  and  $v_B$ ; and 2) a PNC decoder that attempts to decode  $v_A + v_B$  (the sum is bit-wise exclusive-or), referred to as a *coupled* packet. The combined decoding outcomes can be grouped into five events: i) only  $v_A$  is decoded; ii) only  $v_B$  is decoded; iii) only  $v_A + v_B$  is decoded; iv) both  $v_A$  and  $v_B$  are decoded;<sup>1</sup> and v) nothing is decoded. Experiments on the NCMA prototype [3] indicated that all the five events have non-negligible probabilities.

Suppose that each source node has a message formed by  $K$  input packets. The source node A (B) encodes its input packets to a sequence of coded packets  $v_A[i]$  ( $v_B[i]$ ),  $i = 1, \dots, N$  using an erasure-correction code, where  $N$  is the block length of the code. Source nodes A and B transmit packets  $v_A[i]$  and  $v_B[i]$  simultaneously to the sink node. According to the five events above, the outputs of the physical layer of the sink node can be put into three groups. Specifically, for certain subsets  $I_1, I_2, I_3 \subset \{1, 2, \dots, N\}$  with  $(I_1 \cup I_2) \cap I_3 = \emptyset$ , the three groups are

$$\{v_A[i], i \in I_1\}, \{v_B[i], i \in I_2\} \text{ and } \{v_A[i] + v_B[i], i \in I_3\}, \quad (1)$$

where the first group is the coded packets of source node A, the second group is the coded packets of source node B and the third group is the coupled packets.

A natural question that arises is how to encode at the source nodes so that the sink node in NCMA can decode the input packets of all the source nodes reliably using the output packets in (1). In [4], Reed-Solomon codes and uniform random linear codes are used to encode the input packets at the source node. The output packets categorized by the three groups are treated as a coupling of two Reed-Solomon codes (or two uniform random linear codes). The two coupled Reed-Solomon codes (uniform random linear codes) can be decoded jointly by a unified equation system, which is optimal in the sense that as long as there are enough linearly independent equations, the input packets of both source nodes can be decoded [4].

The joint decoding of the coupled Reed-Solomon codes (uniform random linear codes), however, is complex. The decoding complexity by using Gaussian elimination is of  $O((2K)^3 + (2K)^2T)$  finite-field operations, where  $T$  is

<sup>1</sup>If  $v_A$  and  $v_A + v_B$  are decoded, we consider  $v_A$  and  $v_B$  as being decoded since  $v_B = v_A + (v_A + v_B)$ .

the number of field elements in a packet. As a result, the system prototype in [4] can only demonstrate the real-time decoding for low data rates. Further, if NCMA is generalized to accommodate more than two source nodes, the decoding complexity will be much higher. Take an  $L$ -user NCMA system for example, using Reed-Soloman codes (uniform random linear codes) may result in a decoding complexity of  $O(L^3K^3 + L^2K^2T)$  finite-field operations, making real-time decoding even more challenging. This observation motivates us to study a more efficient coding scheme for NCMA with low encoding/decoding complexity.

### B. Paper Contributions

For a general NCMA system with  $L \geq 2$  users, the sink node can decode as many as  $L$  linear combinations with coefficients over a finite field for a set of simultaneously transmitted packets in each timeslot.<sup>2</sup> In this paper, we study how to efficiently recover the original messages of all the users using the linear combinations decoded in different timeslots. This message coding problem induced by NCMA is the channel coding for linear multiple-access channels (MACs), where the output is a set of linear combinations of the multiple input packets.

Fountain codes (e.g., LT codes [9] and Raptor codes [10]) were originally introduced for erasure channels and have the advantages of ratelessness and low encoding/decoding complexity. We propose a digital fountain approach for NCMA, where each user encodes its  $K$  input packets using a fountain code. These linear combinations decoded by the physical layer of the sink node over a number of timeslots are collectively called a *linearly-coupled (LC) fountain code*. We use  $LC-L$  to indicate the LC fountain code involving  $L$  users.

The ordinary BP decoding algorithm of fountain codes is not optimal for LC fountain codes, except for the case of two users. We instead propose a *batched BP decoding* algorithm, which processes the linear combinations decoded from the same timeslot jointly (see Section V-B). The decoding complexity of batched BP decoding is of  $O(LK(\tilde{L}^2 + LT))$  finite-field operations, where  $\tilde{L} \leq L$  is the maximum number of linearly independent combinations that can be decoded by the physical-layer for a single timeslot. The batched BP decoding can be regarded as the combination of local Gaussian elimination and the ordinary BP decoding. We analyze the performance of the batched BP decoding algorithm by performing these two parts iteratively (Theorem 11).

The degree distributions of the original fountain codes designed for the single-user erasure channel is far from optimal for the linear multiple-access channel. We provide a geometric analysis of the convergence of the batched BP decoding (Theorem 15). This convergence analysis induces the optimization problems of the degree distributions of the LC fountain codes. We use binary LC-2 and LC-3 fountain codes to illustrate how to optimize the degree distributions. Since each user has an achievable rate, we formulate two degree distribution optimization problems. The first aims to maximize one user's rate given that the other users' rates are fixed. The second aims to maximize the sum rate of all users. We solve these optimization problems numerically. Our numerical results show that binary LC-2 and LC-3 fountain codes can achieve a rate region close to the capacity region of the linear MAC induced by NCMA.

<sup>2</sup>PNC can also operate over a finite ring [7]. Readers can refer to [7], [8] to see how to use finite rings in PNC and how to extend the results over finite fields to finite rings.

### C. Other Related Works

This paper assumes that the PNC decoder can reliably recover one or more linear combinations of the packets transmitted simultaneously. The decoding of the XOR of the packets of two users has been extensively investigated [11], [12] (see also the overview [13]). The decoding of multiple linear combinations over a larger alphabet has been studied in [2], [7]. Our work in this paper can be applied to NCMA with various PNC schemes.

Zhu and Gastpar [14], [15] recently studied the achievable rate region of Gaussian multiple-access channels by using only a modified compute-and-forward decoder to decode linear combinations of the messages, where the channel gains are known to the transmitters. For a multiple-access channel of  $L$  users, their scheme needs to recover  $L$  linearly independent combinations of the  $L$  users' messages. By contrast, in NCMA, it is not necessary for the physical layer to decode  $L$  linearly independent combinations for each timeslot. The message coding scheme studied in this paper can recover the original messages of all users from the linear combinations decoded in multiple timeslots.

Puducher, Klierer and Fuja [16] studied distributed LT codes for a multiple-access relay network, where the relay node does not receive linear combinations of the packets of the source nodes from the physical layer. They study how to selectively combine the packets received from different source nodes so that the degree distribution observed by the sink node approximates a robust soliton distribution. As [11], [12], Hern and Narayanan [17] also studied PNC for the two-user binary linear MAC, wherein the purpose was to decode the XOR of the packets of the two users. By contrast, for the application of LC fountain codes in NCMA here, we want to recover the input packets of both users.

Another line of works with flavors similar to ours is the study of slotted ALOHA with successive interference cancellation [18]–[23]. In these works, if only one user transmits at a timeslot, the packet can be correctly received; if multiple users transmit at the same time slot, the sink node receives a *collision*, which can be regarded as *one* linear combination of all the packets transmitted. In NCMA, however, the sink node can recover *more than one* independent linear combinations from the collision, so that the essential coding problem is more complicated: in particular, the ordinary BP decoding for erasure channels is not optimal and the ordinary tree-based analysis of BP decoding cannot be directly applied.

## II. PROBLEM FORMULATION

### A. NCMA with Fountain Codes

Fix two positive integers  $L$  and  $T$ . Let  $\Theta$  be an *ordered* set of  $L$  symbols (e.g., A, B, C, and so on). Consider an NCMA system with  $L$  source nodes (users), each of which is labelled by a symbol in  $\Theta$ . Fix a finite field  $\mathbb{F}_q$  of  $q$  elements, called the *base field* and a degree  $m$  extension field  $\mathbb{F}_{q^m}$ . For  $s \in \Theta$ , source node  $s$  has  $K_s$  input packets, called the  $s$ -input packets. All the packets are regarded as column vectors of  $T$  symbols in  $\mathbb{F}_{q^m}$ . Each source node  $s$  encodes its input packets using an LT code with degree distribution  $\Psi_s = (\Psi_s[i], i = 1, \dots, D)$ , where  $D$  is the maximum degree. To encode the  $s$ -input packets, the LT-code encoder first obtains a degree  $d$  by sampling the degree distribution  $\Psi_s$  and then combines  $d$  packets chosen uniformly at random from all the  $s$ -input

packets into a coded packet. The generated packet is called an  $s$ -coded packet. All the  $s$ -coded packets are generated independently.

All the source nodes transmit the coded packets simultaneously using a common wireless channel. Let  $v_s$  be the coded packet transmitted by the source node  $s$ ,  $s \in \Theta$ , in a timeslot. The physical-layer decoder of the sink node tries to decode multiple linear combinations of  $v_s, s \in \Theta$  with coefficients over the base field  $\mathbb{F}_q$ . Suppose that  $B$  linearly independent combinations are decoded ( $B$  may vary from timeslot to timeslot). They can be expressed as

$$[v_s, s \in \Theta]H = [u_1, \dots, u_B], \quad (2)$$

where  $H$  is an  $L \times B$  matrix over  $\mathbb{F}_q$ , called the *transfer matrix*, and  $[v_s, s \in \Theta]$  is the matrix formed by juxtaposing the vectors  $v_s$ , where  $v_{s'}$  comes before  $v_{s''}$  whenever  $s' < s''$ .

Note that in (2), the algebraic operations are over the field  $\mathbb{F}_{q^m}$ . We call the set of packets  $\{u_1, \dots, u_B\}$  decoded in a timeslot a *batch*. We say that the batch is generated by  $\{v_s, s \in \Theta\}$  and packet  $v_s$  is the  $s$ -coded packets embedded in the batch. We assume that each coded packet is only transmitted once. In other words, each coded packet is only embedded in one batch. Different batches may have different generator matrices.

The packets decoded by the physical layer of the sink node from  $N$  timeslots are collectively called an *Linearly-Coupled (LC) fountain code formed by the coupling of  $L$  fountain codes*, or an *LC- $L$  fountain code*, where  $N$  is called the block-length of the code. We assume that the empirical distribution of the transfer matrices converges to  $g$ , i.e., denoting the transfer matrix of the  $i$ -th batch as  $H^{(i)}$ ,

$$\frac{|\{i : 1 \leq i \leq N, H^{(i)} = H\}|}{N} \rightarrow g(H), \quad \text{as } N \text{ tends to infinity,}$$

where the domain of  $g$  is the collection of all the full-column-rank,  $L$ -row matrices over  $\mathbb{F}_q$  (note: this includes all such matrices with  $B$  columns,  $B = 1, \dots, L$ , and an empty matrix when nothing is decoded).

Fix  $0 < \eta_s < 1$ ,  $s \in \Theta$ . For decoding, we try to recover  $\eta_s$  fraction of  $s$ -input packets for each user  $s$ . Precodes can be applied on the original packets of each source node so that recovering a given fraction of the input packets of each source node is sufficient to recover the original input packets [10]. The precodes designed for conventional Raptor codes can be used for our LC fountain codes. Note that the precodes usually operate on the extension field  $\mathbb{F}_{q^m}$ . It is possible to use LC fountain codes without precodes.

In this paper, we focus on three questions:

- 1) How to efficiently decode the LC fountain codes?
- 2) How to analyze the decoding performance?
- 3) How to design the degree distributions?

The general answers to the above questions are given in Section V. Before presenting the general results, we discuss as examples the binary LC-2 fountain code in Section III and the binary LC-3 fountain codes in Section IV.

## B. Performance Bounds

The coding problem described above can be regarded as coding for a linear multiple-access channel (MAC) with  $L$  inputs and one output, where each input is a vector in  $\mathbb{F}_{q^m}^T$  and the output is a sequence of linearly independent

combinations of the input vectors. The relation between the inputs and output is given by (2), where  $H$  is only known for decoding.

Denote by  $\mathcal{H}_L$  the collection of all the full-column-rank,  $L$ -row matrices over  $\mathbb{F}_q$ .  $\mathcal{H}_L$  is the set of all possible transfer matrices of the linear MAC with  $L$  inputs. Let  $\mathbf{H}$  be a random matrix over  $\mathcal{H}_L$ . When all the transfer matrices are independent samples of  $\mathbf{H}$ , we can characterize the capacity region of the linear MAC using the existing result on discrete memoryless MAC [24]. For an  $L$ -row matrix  $H$  and  $S \subset \{1, \dots, L\}$ , denote by  $H^S$  the submatrix of  $H$  formed by the rows indexed by  $S$ . Let  $R_i$  be the rate of the  $i$ -th input in terms of vector per channel use. A rate tuple  $(R_1, \dots, R_L)$  is achievable only if

$$\sum_{i \in S} R_i \leq \mathbb{E}[\text{rk}(\mathbf{H}^S)], \quad \forall S \subset \{1, \dots, L\},$$

where  $\mathbf{H}^S$  is the random matrix defined by

$$\Pr\{\mathbf{H}^S = H'\} = \sum_{H \in \mathcal{H}_L: H^S = H'} \Pr\{\mathbf{H} = H\}.$$

Further, when the empirical distribution of the transfer matrices converges to  $g$ , a rate tuple  $(R_1, \dots, R_L)$  is achievable only if

$$\sum_{i \in S} R_i \leq \sum_{H \in \mathcal{H}_L} g(H) \text{rk}(H^S), \quad \forall S \subset \{1, \dots, L\}.$$

We will evaluate the performance of LC fountain codes and compare their rate regions with the above bound. Define

$$\beta_L = \left( \sum_{H \in \mathcal{H}_L} g(H) \text{rk}(H) \right). \quad (3)$$

The sum rate of all inputs is upper bounded by  $\beta_L$ .

### III. LC-2 FOUNTAIN CODES

In this section, we continue to discuss the two-user NCMA system following Section I-A with the binary field as the base field. Though they are the simplest LC fountain codes, LC-2 fountain codes are non-trivial and of practical interests.

#### A. Parameters

When  $L = 2$ , let  $\Theta = \{A, B\}$  where  $A < B$ . We assume  $q = 2$  here. As mentioned in the introduction, for each timeslot, the nonempty outcome of the physical layer can be grouped into four events corresponding to four transfer matrices

$$H_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, H_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, H_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, H_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4)$$

Suppose that out of the  $N$  batches, transfer matrix  $H_i$  occurs exactly  $g(H_i)N$  times. The total number of output packets decoded by the physical layer in  $N$  timeslots is

$$n = N(g(H_1) + g(H_2) + g(H_3) + 2g(H_4)) = N\beta_2,$$

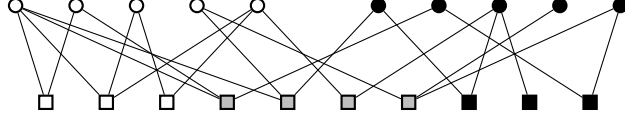


Fig. 1: Linearly-coupled fountain codes. The white/black circles are the A/B-variable nodes, the white/black squares are the A/B-check nodes, and the gray squares are the coupled nodes.

where  $\beta_2$  is defined in (3).

The output packets of an LC-2 fountain code are of two types: *clean* output packets and *coupled* output packets. A output packet is called a *clean* packet if it is an A-coded packet or a B-coded packet. With reference to the definitions in the introduction, the packets in  $\{v_A[i], i \in I_1\}$  and  $\{v_B[i], i \in I_2\}$  are clean output packets. We also simply refer to the clean packets with respect to A and B as A-output packets and B-output packets, respectively. An output packet  $u$  is called a *coupled* output packets if  $u = v_A + v_B$ , where  $v_A$  is an A-coded packet and  $v_B$  is a B-coded packet. The packets in  $\{v_A[i] + v_B[i], i \in I_3\}$  are coupled output packets. The numbers of A-output packets, B-output packets and coupled output packets are  $\alpha_A n$ ,  $\alpha_B n$  and  $\alpha_{A+B} n$ , respectively, where

$$\begin{aligned}\alpha_A &= \frac{g(H_1) + g(H_4)}{\beta_2}, \\ \alpha_B &= \frac{g(H_2) + g(H_4)}{\beta_2}, \\ \alpha_{A+B} &= \frac{g(H_3)}{\beta_2}.\end{aligned}$$

An LC fountain code can be represented by a Tanner graph with the input packets as the variable nodes and the output packets as the check nodes. We also call an input packet a variable node and an output packet a check node henceforth. An example of the Tanner graph is given in Fig. 1.

### B. Ordinary BP Decoding

For LC-2 fountain codes, the (*ordinary*) BP decoding of fountain codes works well, as will be shown. In each step of the decoding algorithm, an output packet of degree one is found, the corresponding input packet is decoded, and the decoded input packet is substituted into the other output packets in which it is involved. The decoding stops when there are no more output packets of degree one. Note that a coupled output packet always has a degree larger than one. Hence, at each step of the BP decoding, only an A or B-output packet of degree one is found and decoded. Suppose that a degree-one A-output packet  $u$  is found at a step of the BP decoding. Then the A-input packet embedded in  $u$  can be recovered. The degrees of the A-output packets and coupled output packets embedding the A-input packet are then reduced by one. The degree reduction of the A-output packets potentially results in new degree-one A-output packets and the degree reduction of the coupled output packets potentially results in new B-output packets, for future steps of the BP decoding.

A check node of degree one is said to be *decodable*. There could be multiple decodable output packets at each step of the BP decoding. We could process the decodable output packets in different orders. But regardless of the

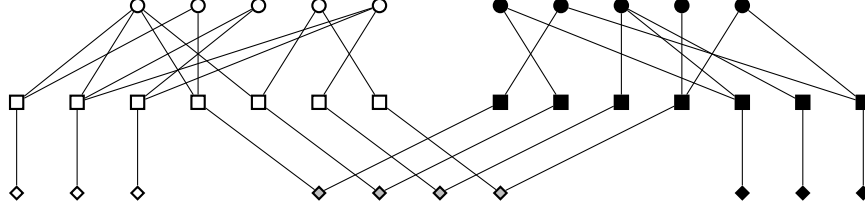


Fig. 2: A three-layer Tanner graph for LC-2 fountain codes. The first layer includes the variable nodes corresponding to the input packets. The second layer includes the check nodes corresponding to the coded packets transmitted by the source nodes. The third layer includes the output packets decoded by NCMA.

processing order, the algorithm will stop with the same remaining output packets. For example, the BP decoding algorithm can process all the decodable output packets in parallel, which is usually described as an *iteration based algorithm*. In each iteration, all the decodable output packets are found and the corresponding input packets are recovered, and then the recovered input packets are substituted into the undecodable output packets. The iteration-based algorithm repeats the above operations until there exist no decodable output packets.

Though it is possible to analyze the BP decoding of LC-2 fountain codes by generalizing the AND-OR tree approach introduced by Luby, Mitzenmacher and Shokrollahi [25], it would be difficult and/or tedious to extend this approach for general LC- $L$  fountain codes  $L > 2$ , where an enhanced BP decoding must be applied to achieved the optimal performance. We provide an approach to analyze LC- $L$  fountain codes based on the existing result of LT codes. Here we introduce the simplified version of this approach for LC-2 fountain codes. Our analysis of LC-2 fountain codes uses the following *round-based BP decoding algorithm*, which has two levels of message passing, illustrated by a three-layer Tanner graph (see Fig. 2). Each round of decoding has two stages. In the first stage, A-check nodes and B-check nodes are decoded separately in the same manner as in conventional LT codes until there are no decodable check nodes left. The coupled nodes are not processed in the first stage. So the decoding in the first stage is equivalent to decoding two LT codes in parallel. The first stage is the message passing between the  $s$ -input packets and  $s$ -output packets for each  $s \in \Theta$ , which can be analyzed using the existing results on LT codes. In the second stage, the coupled nodes are processed by substituting the decoded input packets. This operation lowers the degree of coupled check nodes and may results in new A-check node and B-check node for the next round. The second stage is the message passing between the coupled packets and the decoded input packets, which is the essential technical part for the analysis of LC fountain codes.

### C. Analysis

For degree distributions  $\Psi_s$ ,  $s \in \Theta$ , define

$$\Psi_s(x) = \sum_{i=1}^D \Psi_s[i] x^i \quad \text{and} \quad \Psi'_s(x) = \sum_{i=1}^D i \Psi_s[i] x^{i-1}.$$

We assume that the maximum degree  $D$  does not change with the number of input packets  $K_s$ . This assumption will be justified later by showing that there is a threshold on  $D$  beyond which performance will not be improved.



The following theorem tells us how many input packets are recovered for each source node when the BP decoding stops.

**Theorem 1.** *For each  $s \in \Theta = \{A, B\}$ , fix  $C_s > R_s > 0$ . Consider a sequence of binary LC-2 fountain codes described above with  $K_s/N \leq R_s$ ,  $s \in \Theta$ ,  $N = 1, 2, \dots$ . Define for  $s, s' \in \Theta$  and  $s \neq s'$ ,*

$$F_s(x, y) = F_s(x, y; C_s) = \Psi'_s(x) + \frac{C_s/\beta_2}{\alpha_s + \alpha_{A+B}\Psi_{s'}(y)} \ln(1-x).$$

*Let  $z_s[0] = 0$  and for  $i \geq 1$  let  $z_s[i]$  be the maximum value of  $z$  such that for any  $x \in [0, z]$ , we have*

$$F_s(x, z_{s'}[i-1]) \geq 0,$$

*where  $s' \neq s$ . The sequence  $\{z_s[i]\}$  is increasing and upper bounded. Let  $z_s^*$  be the limit of the sequence  $\{z_s[i]\}$ . Then with probability converging to one, as  $N \rightarrow \infty$ , a BP decoding algorithm stops with at least  $z_s^* K_s$   $s$ -input packets being decoded for all  $s \in \Theta$ .*

*Remark 1.* Consider the round-based BP decoding algorithm. Roughly,  $z_A[i]$  and  $z_B[i]$  in the above theorem are the fractions of the decoded A-input packets and B-input packets after the  $i$ -th round BP decoding.

*Sketch of the proof:* The theorem will be proved as a special case of Theorem 11 to be presented later. Here we give a sketch of the proof. Recall an existing result of LT codes [10]. Fix  $C' > R' > 0$ . Consider an LT code with  $K$  input packets,  $n' \geq K/R'$  output packets and degree distribution  $\Psi(x)$ . If for some  $0 < z < 1$  we have

$$\Psi'(x) + C' \ln(1-x) \geq 0, \forall x \in [0, z],$$

then the code can recover at least  $zK$  input packets with high probability when  $n'$  is sufficiently large.

Consider the round-based BP decoding algorithm introduced in the last subsection. In each round, two LT codes are decoded in parallel. We outline the analysis of the first two rounds. Taking source node A for example, in the first stage of the first round of decoding, the number of A-input packets is  $K_A$  and the number of A-output packets is  $\alpha_A n$ . By the aforementioned result of LT codes, we know that with high probability at least  $z_A[1]K_A$  A-input packets can be recovered at the end of the first round when  $n$  is large.

In the second stage of the first round, the decoded input packets are substituted into the coupled packets. Consider a coupled output packet  $u = v_A + v_B$ , where  $v_A$  ( $v_B$ ) is an A-coded (B-coded) packet. Packet  $v_A$  can be recovered after substitution as long as  $v_B$  is a linear combination of the decoded B-input packets. Since the set of B-input packets embedded in  $v_B$  is chosen uniformly, the probability that  $v_B$  is resolved after the first stage is at least

$$\sum_d \Psi_B[d] \frac{\binom{z_B[1]K_B}{d}}{\binom{K_B}{d}} \approx \Psi_B(z_B[1]).$$

That is, the probability that  $v_A$  can be recovered (as an A-output packet) in the BP decoding in the second round is at least  $\Psi_B(z_B[1])$ . Similarly, the probability that  $v_B$  can be recovered in the BP decoding in the second round is at least  $\Psi_A(z_A[1])$ .

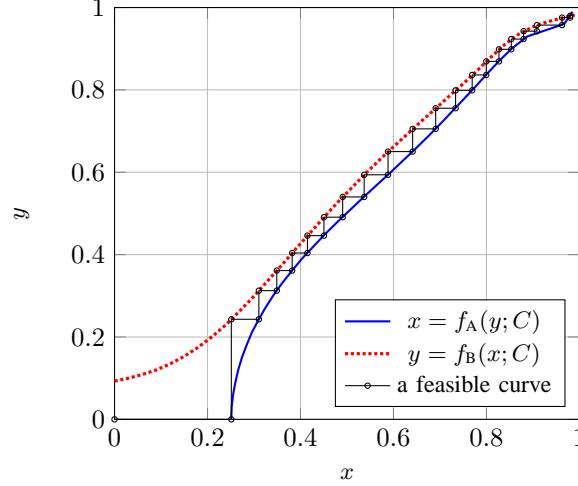


Fig. 3: Curves  $x = f_A(y)$  and  $y = f_B(x)$  with  $\alpha_A = \alpha_B = 0.25$  and  $\alpha_{A+B} = 0.5$ . The first intersection is  $(0.98, 0.98)$ .

In the second round, the total number of A-output packets is at least  $n[\alpha_A + \alpha_{A+B}\Psi_B(z_B[1])]$ , and these output packets along with the  $K_A$  A-input packets form an LT code. Using again the result of LT codes, we know that at least  $z_A[2]K_A$  A-input packets can be recovered at the end of the second round. ■

Let us give a more explicit characterization of the limits  $(z_A^*, z_B^*)$ . Define

$$f_A(y; C_A) = \max \{z : F_A(x, y; C_A) \geq 0, \forall x \in [0, z]\},$$

$$f_B(x; C_B) = \max \{z : F_B(y, x; C_B) \geq 0, \forall y \in [0, z]\}.$$

We also write  $f_A(y; C_A)$  and  $f_B(x; C_B)$  as  $f_A(y)$  and  $f_B(x)$ , respectively, when  $C_A$  and  $C_B$  are implied by the context. Both  $f_A(y)$  and  $f_B(x)$  are increasing. The two sequences in Theorem 1 satisfy  $z_A[i] = f_A(z_B[i-1])$  and  $z_B[i] = f_B(z_A[i-1])$  for  $i \geq 1$ .

The following lemma gives a geometric characterization of the limits of the sequences  $\{z_A[i]\}$  and  $\{z_B[i]\}$ .

**Lemma 2.** *The limit point  $(z_A^*, z_B^*)$  of the two sequences defined in Theorem 1 for LC-2 fountain codes is the first intersection of the curve  $x = f_A(y)$  and the curve  $y = f_B(x)$ ,  $x, y \in [0, 1]$ .*

*Proof:* The lemma can be proved using the monotonicity of  $f_A$  and  $f_B$  and is a special case of Lemma 14. ■

Fig. 3 illustrates a pair of functions  $f_A$  and  $f_B$ . For a pair  $(a, b)$  in the region  $\{(x, y) : 0 \leq x, y \leq 1\}$ , we say  $(a, b)$  is  $(C_A, C_B)$ -feasible for an LC-2 fountain code if  $a \leq f_A(b; C_A)$  and  $b \leq f_B(a; C_B)$ . A curve is  $(C_A, C_B)$ -feasible for an LC-2 fountain code if every point on the curve is  $(C_A, C_B)$ -feasible. A point/curve is said to be *feasible* when  $C_A$  and  $C_B$  are implied. One property of the feasible points is that if both  $(c, d)$  and  $(c, d')$  are feasible, then the vertical segment between these two points is feasible. This is because for any  $y \in [d', d]$  (assuming  $d' \leq d$ ), we have  $y \leq d \leq f_B(c)$  and  $c \leq f_A(d') \leq f_A(y)$  (since  $f_A$  is an increasing function). The same property holds for horizontal line segments. For example, the zig-zag curve in Fig. 3 is a feasible curve.

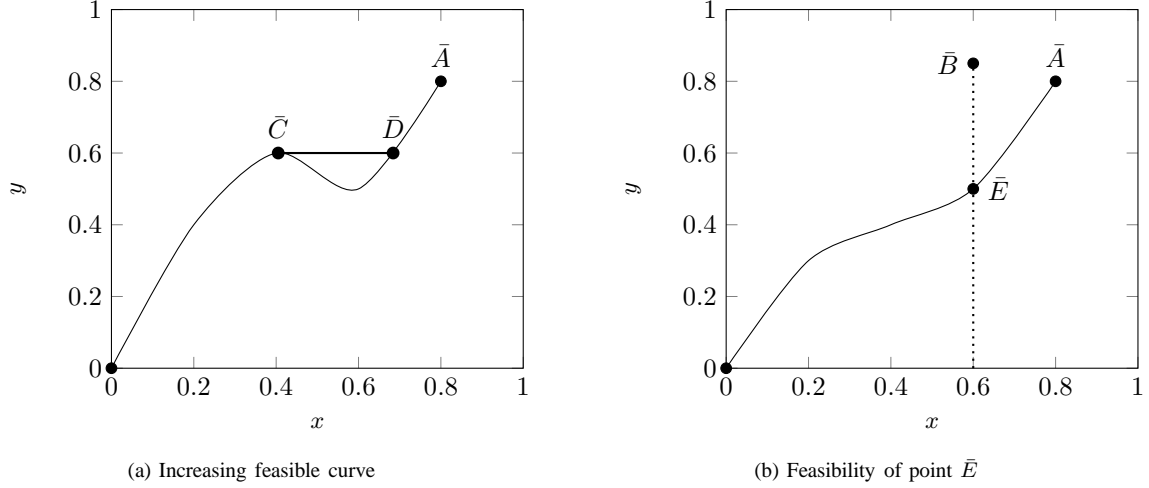


Fig. 4: An illustration of the proof of Theorem 3.

**Theorem 3.** For each  $s \in \Theta = \{A, B\}$ , fix  $C_s > R_s > 0$ . Consider a sequence of binary LC-2 fountain codes with  $N = 1, 2, \dots$ , where  $K_s/N \leq R_s$  for  $s \in \Theta$ . For any pair  $(a_A, a_B)$ , if there exists a  $(C_A, C_B)$ -feasible continuous curve  $(x(t), y(t))$  between the origin and  $(a_A, a_B)$ , then i) a BP decoding algorithm will stop with at least  $a_s K_s$   $s$ -input packets being decoded for all  $s \in \Theta$  with probability converging to one when  $N \rightarrow \infty$ , and ii) there exists an increasing, continuous and  $(C_A, C_B)$ -feasible curve  $(\tilde{x}(t), \tilde{y}(t))$  between the origin and  $(a_A, a_B)$ .

*Sketch of the proof:* The theorem will be proved as a special case of Theorem 15. Here we give a sketch of the proof. Fig. 4 illustrates the main ideas, in which the point  $(a_A, a_B)$  is labeled by  $\bar{A}$ . We first show the second claim. Suppose there exists a feasible curve from the origin to point  $\bar{A}$ , which is not increasing, e.g., the thin solid curve in Fig. 4a. Point  $\bar{C}$  is a local maximum of the curve and point  $\bar{D}$  is also on the curve which share the same  $y$ -coordinate as point  $\bar{C}$ . We can replace the part of the curve between points  $\bar{C}$  and  $\bar{D}$  by the line segment (the thick solid line segment in the figure) between points  $\bar{C}$  and  $\bar{D}$ . The new curve is increasing. The points on the line segment between points  $\bar{C}$  and  $\bar{D}$  are feasible since both  $\bar{C}$  and  $\bar{D}$  are feasible. The second claim in the theorem can be proved by repeating the above procedure.

It is sufficient to prove the first claim for increasing curve  $(x(t), y(t))$ . Fix  $C'_A$  and  $C'_B$  such that  $R_A < C'_A < C_A$  and  $R_B < C'_B < C_B$ . Denote by  $\bar{B} = (b_A, b_B)$  the first intersection of curves  $x = f_A(y; C'_A)$  and  $y = f_B(x; C'_B)$ . If both  $b_A \geq a_A$  and  $b_B \geq a_B$ , the first claim holds by Lemma 2 and Theorem 1. We then show by contradiction that it is not possible that either  $b_A < a_A$  or  $b_B < a_B$ . Suppose  $b_A < a_A$  and  $b_B \geq a_B$  as illustrated in Fig. 4b. Consider the point  $\bar{E} = (b_A, e_B)$  on the curve  $(x(t), y(t))$ , where  $e_B \leq a_B \leq b_B$ . The contradiction is that  $\bar{E}$  is not  $(C_A, C_B)$ -feasible since

$$b_A = f_A(b_B, C') \geq f_A(e_B, C') > f_A(e_B, C),$$

where the inequalities follow from the monotonicity of the function  $f_A$ . ■

#### D. Optimizations

Given the parameters  $\alpha_A$ ,  $\alpha_B$  and  $\alpha_{A+B}$ , we want to design a binary LC-2 fountain codes such that at least  $\eta_A$  fraction of A-input packets and  $\eta_B$  fraction of B-input packets can be decoded by BP decoding. By Theorem 3, a rate pair  $(\eta_A C_A, \eta_B C_B)$  is *achievable* by BP decoding if there exists a  $(C_A, C_B)$ -feasible curve between the origin and  $(\eta_A, \eta_B)$ . Theorem 3 also enables us to consider only the increasing curves from the origin to  $(\eta_A, \eta_B)$ .

By definition, a point  $(\hat{x}, \hat{y})$  is  $(C_A, C_B)$ -feasible if  $\hat{x} \leq f_A(\hat{y}; C_A)$  and  $\hat{y} \leq f_B(\hat{x}; C_B)$ , which are equivalent to

$$F_A(x, \hat{y}; C_A) \geq 0, \quad \forall x \in [0, \hat{x}],$$

$$F_B(y, \hat{x}; C_B) \geq 0, \quad \forall y \in [0, \hat{y}],$$

that is,

$$[\alpha_A + \alpha_{A+B} \Psi_B(\hat{y})] \Psi'_A(x) + C_A/\beta_2 \ln(1-x) \geq 0, \quad \forall x \in [0, \hat{x}], \quad (5)$$

$$[\alpha_B + \alpha_{A+B} \Psi_A(\hat{x})] \Psi'_B(y) + C_B/\beta_2 \ln(1-y) \geq 0, \quad \forall y \in [0, \hat{y}]. \quad (6)$$

We only evaluate the zig-zag type of curves (see Fig. 3 for an example). Fix a positive integer  $t_{\max}$  and two sequences of real numbers  $x_t, y_t, t = 0, 1, \dots, t_{\max}$  with

$$0 = x_0 \leq x_1 \leq \dots \leq x_{t_{\max}} = \eta_A,$$

$$0 = y_0 \leq y_1 \leq \dots \leq y_{t_{\max}} = \eta_B.$$

The curve formed by line segments  $(x_t, y_t) - (x_{t+1}, y_t) - (x_{t+1}, y_{t+1}), t = 0, 1, \dots, t_{\max} - 1$  is an increasing zig-zag curve from the origin to  $(\eta_A, \eta_B)$ . As explained before, the vertical (horizontal) line segment between two feasible points is feasible. So we only need to check the feasibility of the points

$$(x_0, y_0), (x_1, y_0), (x_1, y_1), (x_2, y_1), \dots, (x_{t_{\max}}, y_{t_{\max}}). \quad (7)$$

We do not lose optimality since all increasing curves can be approximated closely by such zig-zag curves when  $t_{\max}$  is sufficiently large.

Now we are ready to introduce the optimization problems for binary LC-2 fountain codes. Since we have a pair of coding rates, we may fix one and maximize the other or maximize the sum rate. Fix  $t_{\max}, C_B, \eta_A$  and  $\eta_B$ . The following optimization problem maximizes the achievable rate of source node A for a given rate of source node B:

$$\max \eta_A \theta_A \beta_2$$

$$\text{s.t. } x_0 = 0, y_0 = 0, x_{t_{\max}} = \eta_A, y_{t_{\max}} = \eta_B,$$

$$\forall t = 1, \dots, t_{\max}, \quad x_t \geq x_{t-1}, y_t \geq y_{t-1}, \quad (8)$$

$$[\alpha_A + \alpha_{A+B} \Psi_B(y_{t-1})] \Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (x_{t-1}, x_t],$$

$$[\alpha_B + \alpha_{A+B} \Psi_A(x_t)] \Psi'_B(y) + C_B/\beta_2 \ln(1-y) \geq 0, \quad \forall y \in (y_{t-1}, y_t],$$

where the variables of the optimization are  $\theta_A, x_t, y_t, t = 1, \dots, t_{\max}, \Psi_A$  and  $\Psi_B$ . Note that in the above optimization, we do not require the inequalities in the last two lines to be satisfied for  $x$  or  $y$  starting from zero as

in (5) and (6). But the last two lines can still guarantee that the points in (7) are all feasible due to the following property. Suppose that for  $i = 1, \dots, t$  we have

$$[\alpha_B + \alpha_{A+B} \Psi_A(x_i)] \Psi'_B(y) + C_B/\beta_2 \ln(1-y) \geq 0, \quad \forall y \in (y_{i-1}, y_i].$$

Due to the monotonic property of  $\Psi_A(x)$  and  $x_t \geq x_i$  for  $i < t$ , we have for  $i = 1, \dots, t$

$$[\alpha_B + \alpha_{A+B} \Psi_A(x_t)] \Psi'_B(y) + C_B/\beta_2 \ln(1-y) \geq 0, \quad \forall y \in (y_{i-1}, y_i].$$

Combining the  $t$  equalities, we have

$$[\alpha_B + \alpha_{A+B} \Psi_A(x_t)] \Psi'_B(y) + C_B/\beta_2 \ln(1-y) \geq 0, \quad \forall y \in (0, y_t].$$

Similarly, the second last line in the above optimization implies

$$[\alpha_A + \alpha_{A+B} \Psi_B(y_{t-1})] \Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (0, x_t].$$

We can also write an optimization to maximize the rate of the source node B.

For given  $t_{\max}$ ,  $\eta_A$  and  $\eta_B$ , we can maximize the sum rate of both source nodes as follows:

$$\begin{aligned} & \max \beta_2(\eta_A \theta_A + \eta_B \theta_B) \\ & \text{s.t. } x_0 = 0, y_0 = 0, x_{t_{\max}} = \eta_A, y_{t_{\max}} = \eta_B, \\ & \quad \forall t = 1, \dots, t_{\max}, \quad x_t \geq x_{t-1}, y_t \geq y_{t-1}, \\ & \quad [\alpha_A + \alpha_{A+B} \Psi_B(y_{t-1})] \Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (x_{t-1}, x_t], \\ & \quad [\alpha_B + \alpha_{A+B} \Psi_A(x_t)] \Psi'_B(y) + \theta_B \ln(1-y) \geq 0, \quad \forall y \in (y_{t-1}, y_t], \end{aligned} \tag{9}$$

where the variables of the optimization are  $\theta_A$ ,  $\theta_B$ ,  $x_t$ ,  $y_t$ ,  $t = 1, \dots, t_{\max}$ , degree distributions  $\Psi_A$  and  $\Psi_B$ .

The maximum degree  $D$  can be similarly bounded as for fountain codes.

**Lemma 4.** Consider optimizations (8) and (9). For  $s \in \{A, B\}$ , using degrees larger than  $\lceil 1/(1-\eta_s) \rceil - 1$  for  $\Psi_s$  does not give better optimal values.

*Proof:* We use problem (9) as an example to prove the lemma. Consider an integer  $\Delta$  such that  $1 - \eta_A \geq \frac{1}{\Delta+1}$ . Let  $\Psi_A$  be a degree distribution with  $\sum_{d>\Delta} \Psi_A[d] > 0$ . Construct a new degree distribution  $\tilde{\Psi}_A$  with  $\tilde{\Psi}_A[d] = \Psi_A[d]$  for  $d < \Delta$ ,  $\tilde{\Psi}_A[\Delta] = \sum_{d \geq \Delta} \Psi_A[d]$  and  $\tilde{\Psi}_A[d] = 0$  for  $d > \Delta$ . We have  $\tilde{\Psi}_A(x) - \Psi_A(x) = \sum_{d>\Delta} \Psi_A[d](x^\Delta - x^d) > 0$  and

$$\tilde{\Psi}'_A(x) - \Psi'_A(x) = \sum_{d>\Delta} \Psi_A[d](\Delta x^{\Delta-1} - dx^{d-1}).$$

Since for  $d \geq \Delta$

$$\frac{(d+1)x^d}{dx^{d-1}} = \frac{d+1}{d}x \leq \frac{d+1}{d}\eta_A \leq \frac{\Delta+1}{\Delta}\eta = 1,$$

we have  $\tilde{\Psi}'_A(x) \geq \Psi'_A(x)$ . Thus,  $\tilde{\Psi}_A$  does not give worse optimal value than  $\Psi_A$ . The part of the lemma for  $\Psi_B$  can be similarly proved. ■

TABLE I

ACHIEVABLE RATES OF BINARY LC-2 FOUNTAIN CODES FOR  $\eta_A = \eta_B = 0.98$ . IN BOTH (8) AND (9), THE OBJECTIVE FUNCTIONS ARE MODIFIED BY REMOVING  $\beta_2$ .  $\hat{R}_A/\beta_2$  IS OBTAINED BY SOLVING (8) WITH  $C_B/\beta_2 = \alpha_B/\eta_B$ , AND  $\hat{R}_{\text{sum}}/\beta_2$  IS OBTAINED BY SOLVING (9).

$\alpha_{A+B}$	$\alpha_A$	$\alpha_B$	$\hat{R}_A/\beta_2$	$\hat{R}_{\text{sum}}/\beta_2$
0.05	0.475	0.475	0.5135	0.9879
0.25	0.375	0.375	0.5962	0.9797
	0.45	0.3	0.6701	0.9823
0.5	0.25	0.25	0.7022	0.9617
	0.375	0.125	0.8292	0.9724
	0.45	0.05	0.9090	0.9616
0.75	0.125	0.125	0.8137	0.9510
	0.1875	0.0625	0.8854	0.9571
	0.225	0.025	0.9359	0.9589
0.95	0.025	0.025	0.9317	0.9496

### E. Achievable Rates

Given the distribution  $g$  of the transfer matrix, we know from Section II-B that a rate pair  $(R_A, R_B)$  is achievable only if

$$\begin{aligned}
 R_A &\leq \sum_i g(H_i) \text{rk}(H_i^{\{A\}}) = g(H_1) + g(H_3) + g(H_4) = \beta_2(\alpha_A + \alpha_{A+B}), \\
 R_B &\leq \sum_i g(H_i) \text{rk}(H_i^{\{B\}}) = g(H_2) + g(H_3) + g(H_4) = \beta_2(\alpha_B + \alpha_{A+B}), \\
 R_A + R_B &\leq \sum_i g(H_i) \text{rk}(H_i) = \beta_2.
 \end{aligned}$$

Instead of specifying a value of  $\beta_2$ , we remove  $\beta_2$  from the objective functions of both (8) and (9) so that the optimal values are the normalized (sum) rates. The best numerical results obtained by evaluating the modified optimization (9) are listed in Table I, where we can see that the normalized achievable sum rates are all close 1, the upper bound. One of the vertex of the above region is  $R_A = \beta_2(\alpha_A + \alpha_{A+B})$  and  $R_B = \beta_2\alpha_B$ . We evaluate (8) with  $C_B/\beta_2 = \alpha_B/\eta_B$ . From Table I, readers can verify that the normalized achievable rates of user A are all close to the corresponding values of  $\alpha_A + \alpha_{A+B}$ . Note that for the values obtained in Table I,  $\beta_2$  can be any value in the range  $(0, 2)$ .

The optimizations (8) and (9) are non-convex and hence we may not obtain the globally optimal values. We discuss in the appendix how to solve these optimizations. Nevertheless, the numerical results show that the obtained suboptimal rates are all very close to the bound we provided above. Since the values may not be globally optimal, for each row it is possible that the value of  $\alpha_B$  plus the value in the second last column is larger than the value in the last column.

#### IV. LC-3 FOUNTAIN CODES

Our discussion of LC-2 function codes can be generalized to LC- $L$  with  $L > 2$ . However, the generalization involves new features absent in the LC-2 case. In this section, we use the LC-3 fountain codes to illustrate the implications of these new features for the design and analysis of general LC- $L$  fountain codes.

##### A. Batches

For  $L = 3$ , let  $\Theta = \{A, B, C\}$ , where  $A < B < C$ . We assume  $q = 2$  here. Compared with LC-2 fountain codes, we have a new type of coupled packet  $v_A + v_B + v_C$  embedded with three (rather than just two) coded packets, where  $v_s$ ,  $s \in \Theta$  is transmitted by source node  $s$ . We say an output packet of a batch is *autonomous* if none of the coded packets embedded in it is embedded in other output packets of the batch. For example, if the physical layer decodes  $v_A$  and  $v_A + v_B + v_C$ , we get two non-autonomous output packets. But we can transform them into autonomous output packets by reducing  $v_A + v_B + v_C$  to  $v_B + v_C$ . On the other hand, if the physical layer decodes  $v_A + v_B$  and  $v_B + v_C$ , we cannot transform them into autonomous output packets.

For each timeslot, if the physical layer decodes only one packet, the packet is autonomous. If the physical layer decodes three linearly independent packets, after linear transformation, this is equivalent to obtaining three autonomous output packets  $v_A$ ,  $v_B$  and  $v_C$ . If the physical layer decodes two linearly independent packets, it is possible to have non-autonomous output packets as seen in the above example. For an LC-3 fountain code, all non-autonomous output packets can be put into the form  $\{v_A + v_C, v_B + v_C\}$  after linear transformation. We will see that to achieve optimal performance, non-autonomous output packets should be handled in a different way from how autonomous output packets are handled.

The combined decoding outcomes of the physical layer, after proper linear transformation, can be categorized into the following eight cases:

- 1) Only  $v_s$  is decoded, where  $s \in \Theta$ . The corresponding transfer matrix is one of the following:

$$H_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, H_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, H_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

- 2) Only  $v_s$  and  $v_{s'}$  are decoded, where  $s < s' \in \Theta$ . The corresponding transfer matrix is one of the following:

$$H_4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, H_5 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, H_6 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

- 3) All the three packets  $v_A$ ,  $v_B$  and  $v_C$  are decoded. The corresponding transfer matrix is

$$H_7 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

4) Only  $v_s + v_{s'}$  is decoded, where  $s < s' \in \Theta$ . The corresponding transfer matrix is one of the following:

$$H_8 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, H_9 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, H_{10} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

5) Only  $v_A + v_B + v_C$  is decoded. The corresponding transfer matrix is

$$H_{11} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

6) Only  $v_s + v_{s'}$  and  $v_{s''}$  are decoded, where  $s \neq s' \neq s'' \in \Theta$  and  $s < s'$ . The corresponding transfer matrix is one of the following:

$$H_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, H_{13} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, H_{14} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

7) Two non-autonomous output packets are decoded. The corresponding transfer matrix is one of the following:

$$H_{15} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, H_{16} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, H_{17} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

8) Nothing is decoded.

Suppose that the number of batches with the transfer matrix  $H_i$  occurring is exactly  $g(H_i)N$ . The total number of output packets is  $n = \beta_3 N$ , where  $\beta_3$  is defined in (3).

An autonomous output packet of the form  $\sum_{s \in S} v_s$  for certain  $S \subset \Theta$  is called an *S-output packet*. Define for LC-3 fountain codes

$$\begin{aligned} \alpha_A &= \frac{g(H_1) + g(H_4) + g(H_5) + g(H_7) + g(H_{13})}{\beta_3}, \\ \alpha_B &= \frac{g(H_2) + g(H_4) + g(H_6) + g(H_7) + g(H_{12})}{\beta_3}, \\ \alpha_C &= \frac{g(H_3) + g(H_5) + g(H_6) + g(H_7) + g(H_{14})}{\beta_3}, \\ \alpha_{A+B} &= \frac{g(H_8) + g(H_{14})}{\beta_3}, \\ \alpha_{A+C} &= \frac{g(H_9) + g(H_{12})}{\beta_3}, \\ \alpha_{B+C} &= \frac{g(H_{10}) + g(H_{13})}{\beta_3}, \\ \alpha_{A+B+C} &= \frac{g(H_{11})}{\beta_3}, \\ \bar{\alpha}_A &= \frac{g(H_{16})}{\beta_3}, \end{aligned}$$



$$\bar{\alpha}_B = \frac{g(H_{17})}{\beta_3},$$

$$\bar{\alpha}_C = \frac{g(H_{15})}{\beta_3}.$$

For  $s \neq s' \neq s''$ , we also write  $\alpha_s = \alpha_{\{s\}}$ ,  $\alpha_{s+s'} = \alpha_{\{s,s'\}}$  and  $\alpha_{s+s'+s''} = \alpha_{\{s,s',s''\}}$ . We have

$$\sum_{S \subset \Theta: |S| \geq 1} \alpha_S + 2 \sum_{s \in \Theta} \bar{\alpha}_s = 1.$$

For each  $S \subset \Theta$  and  $S \neq \emptyset$ , the number of (autonomous)  $S$ -output packets is  $\alpha_S n$ . When  $S = \{s\}$ , an  $S$ -output packet is an  $s$ -output packet. Totally, we have  $n \sum_{S \subset \Theta: |S| \geq 1} \alpha_S$  autonomous output packets. Let

$$\bar{\alpha} = \bar{\alpha}_A + \bar{\alpha}_B + \bar{\alpha}_C.$$

The remaining  $n(1 - \sum_{S \subset \Theta: |S| \geq 1} \alpha_S) = 2n\bar{\alpha}$  output packets are non-autonomous output packets contained in  $n\bar{\alpha} = N[g(H_{15}) + g(H_{16}) + g(H_{17})]$  batches.

### B. Batched BP Decoding

The ordinary BP decoding of fountain codes can be used to decode LC-3 fountain codes. But as we will show in the next example, we can improve the decoding performance by exploiting the batch structure of the non-autonomous output packets in the decoding process.

Consider a batch of two non-autonomous output packets  $u_1 = v_A + v_B$  and  $u_2 = v_B + v_C$  (see the illustration in Fig. 5). Suppose that when the ordinary BP decoding stops, packet  $v_A$  is a linear combination of the already-decoded A-input packets, packet  $v_B$  has a degree larger than one, and packet  $v_C$  has degree one. The ordinary BP decoding substitutes the already-decoded A-input packets in  $u_1$  and recovers  $v_B$ . But since only already-decoded input packets can be substituted, the ordinary BP decoding does not substitute  $v_B$  into  $u_2$  to recover  $v_C$ , and hence the BP decoding cannot be resumed. However, if we allow joint processing of  $u_1$  and  $u_2$ , we can substitute  $v_B$  into  $u_2$  to obtain  $v_C$  and hence the BP decoding can be resumed since  $v_C$  has degree one.

Motivated by the above example, we propose the *batched BP decoding* for LC-3 codes. Recall that only batches with transfer matrices  $H_{15}, H_{16}$  and  $H_{17}$  have non-autonomous output packets. The batched BP decoding is the same as the ordinary BP decoding except that it also solves the linear systems of equations (at the second stage of each round):

$$[u_1, u_2] = [v_A, v_B, v_C]H_{15}, \quad (10)$$

where  $u_1$  and  $u_2$  are the two output packets of the batch. Note that for batches with transfer matrices  $H_{16}$  and  $H_{17}$ , the associated linear systems are equivalent to (10). When any one of  $v_A, v_B$  or  $v_C$  is the linear combination of the already-decoded input packets, the batched BP decoding solves (10) to resolve the value of the other two.

### C. Analysis

The following theorem tells us how many input packets are recovered for each source node when the *ordinary* BP decoding stops for binary LC-3 fountain codes.

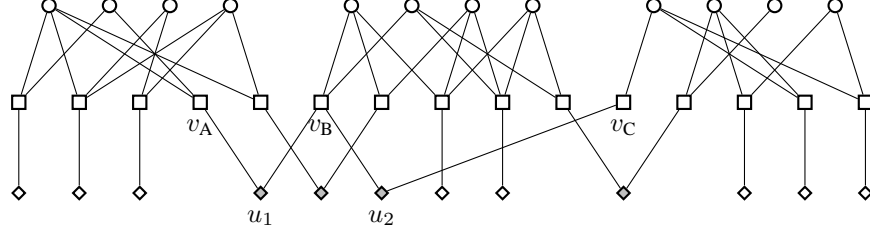


Fig. 5: A three-layer Tanner graph for LC-3 fountain codes. The first layer includes the variable nodes corresponding to the input packets. The second layer includes the check nodes corresponding to the coded packets transmitted by the source nodes. The third layer includes the output packets decoded by NCMA. In this graph,  $u_1$  and  $u_2$  forms a batch with two non-autonomous packets.

**Theorem 5.** For each  $s \in \Theta = \{A, B, C\}$ , fix  $C_s > R_s > 0$  and consider a sequence of binary LC-3 fountain codes described above with  $K_s/N \leq R_s$ ,  $s \in \Theta$ ,  $N = 1, 2, \dots$ . For  $s \neq s' \neq s'' \in \Theta$ , define

$$F_s^o(x, x', x'') = \Psi_s'(x) + \frac{C_s/\beta_3}{\alpha_s + \lambda_1(s) + \lambda_2^o(s)} \ln(1-x),$$

where

$$\begin{aligned} \lambda_1(s) &= \alpha_{s+s'} \Psi_{s'}(x') + \alpha_{s+s''} \Psi_{s''}(x'') + \alpha_{s+s'+s''} \Psi_{s'}(x') \Psi_{s''}(x''), \\ \lambda_2^o(s) &= \bar{\alpha}_s (\Psi_{s'}(x') + \Psi_{s''}(x'') - \Psi_{s'}(x') \Psi_{s''}(x'')) + \bar{\alpha}_{s'} \Psi_{s'}(x') + \bar{\alpha}_{s''} \Psi_{s''}(x''). \end{aligned}$$

Let  $z_s^o[0] = 0$ , and for  $i \geq 1$  let  $z_s^o[i]$  be the maximum value of  $z$  such that for any  $x \in [0, z]$ , we have

$$F_s^o(x, z_{s'}^o[i-1], z_{s''}^o[i-1]) \geq 0,$$

where  $s \neq s' \neq s''$  and  $s' < s''$ . The sequence  $\{z_s^o[i]\}$  is increasing and upper bounded. Let  $z_s^{\circledast}$  be the limit of the sequence  $\{z_s^o[i]\}$ . Then with probability converging to one, as  $N \rightarrow \infty$ , the ordinary BP decoding algorithm stops with at least  $z_s^{\circledast} K_s$   $s$ -input packets being decoded for all  $s \in \Theta$ .

*Proof:* The theorem will be proved as a special case of Theorem 11. ■

The following theorem tells us how many input packets are recovered for each source node when the *batched* BP decoding stops for binary LC-3 fountain codes.

**Theorem 6.** For each  $s \in \Theta = \{A, B, C\}$ , fix  $C_s > R_s > 0$  and consider a sequence of binary LC-3 fountain codes described above with  $K_s/N \leq R_s$ ,  $s \in \Theta$ ,  $N = 1, 2, \dots$ . For  $s \neq s' \neq s'' \in \Theta$ , define

$$F_s(x, x', x'') = \Psi_s'(x) + \frac{C_s/\beta_3}{\alpha_s + \lambda_1(s) + \lambda_2(s)} \ln(1-x),$$

where

$$\begin{aligned} \lambda_1(s) &= \alpha_{s+s'} \Psi_{s'}(x') + \alpha_{s+s''} \Psi_{s''}(x'') + \alpha_{s+s'+s''} \Psi_{s'}(x') \Psi_{s''}(x''), \\ \lambda_2(s) &= \bar{\alpha} (\Psi_{s'}(x') + \Psi_{s''}(x'') - \Psi_{s'}(x') \Psi_{s''}(x'')). \end{aligned}$$

Let  $z_s[0] = 0$  and for  $i \geq 1$  let  $z_s[i]$  be the maximum value of  $z$  such that for any  $x \in [0, z]$ , we have

$$F_s(x, z_{s'}[i-1], z_{s''}[i-1]) \geq 0,$$

where  $s \neq s' \neq s''$  and  $s' < s''$ . The sequence  $\{z_s[i]\}$  is increasing and upper bounded. Let  $z_s^*$  be the limit of the sequence  $\{z_s[i]\}$ . Then with probability converging to one, as  $N \rightarrow \infty$ , the batched BP decoding algorithm stops with at least  $z_s^* K_s$   $s$ -input packets being decoded for all  $s \in \Theta$ .

*Remark 2.* The performance of the batched BP decoding characterized in the above theorem does not depend on the individual values of  $\bar{\alpha}_A, \bar{\alpha}_B, \bar{\alpha}_C$  as long as their summation is the same.

*Remark 3.* In the above two theorems,  $\lambda_2^o(s) \leq \lambda_2(s)$  for all  $s$  and the inequalities are strict for at least 2 users. Therefore, in general,  $z_s^{\oplus} \leq z_s^*$  for all  $s$  and the inequalities are strict for at least two users.

*Sketch of the proof:* The theorem will be proved as a special case of Theorem 11 to be presented later. Here we give a sketch of the proof. Compared with Theorem 1, the major difference is the denominator of the second term of  $F_s$ . So we focus on how the denominator is obtained in this sketch. The first stage of the batched BP decoding is similar to that of binary LC-2 fountain codes so we consider the second stage of the first round in the following. Compared with the LC-2 fountain codes, we have more types of couples packets and non-autonomous output packets for LC-3 fountain codes.

Consider an output packet  $u = v_A + v_B + v_C$ , where  $v_s$  is an  $s$ -coded packet. Packet  $v_A$  can be recovered as long as both  $v_B$  and  $v_C$  are linear combinations of the decoded input packets at the first stage. So at the second stage of the first round, the probability that  $v_A$  can be recovered is at least  $\Psi_B(z_B[1])\Psi_C(z_C[1])$ .

Consider a batch formed by transfer matrix  $H_{15}$  and coded packets  $v_A, v_B$  and  $v_C$ . If either  $v_B$  or  $v_C$  is a linear combination of the decoded input packets at the first stage,  $v_A$  can be recovered and used in the BP decoding in the next round. So at the second stage of the first round, the probability that  $v_A$  can be recovered by solving (10) is at least  $1 - (1 - \Psi_B(z_B[1]))(1 - \Psi_C(z_C[1]))$ .

Counting all coupled  $S$ -output packets with  $A \in S$  and all the batches with transfer matrices  $H_{15}, H_{16}$  and  $H_{17}$ , we get that the number of  $A$ -output packets recovered is at least  $n[\alpha_A + \lambda_1(A) + \lambda_2(A)]$  at the second stage of the first round. ■

For  $s \neq s' \neq s'' \in \Theta$  with  $s' < s''$ ,  $F_s$  defined in Theorem 6 can be rewritten as

$$F_s(x, x', x''; C_s) = \Psi'_s(x) + \frac{C_s/\beta_3}{\Sigma(\Psi_{s'}(x'), \Psi_{s''}(x''))} \ln(1 - x),$$

where

$$\Sigma(y, z) = \alpha_s + \alpha_{s+s'}y + \alpha_{s+s''}z + \alpha_{\Theta}yz + \bar{\alpha}(y + z - yz).$$

Fixing one of the variables,  $\Sigma(y, z)$  is an increasing function of the other variable. For  $s \in \Theta$ , define

$$f_s(x', x'') = f_s(x', x''; C_s) = \max \{z : F_s(x, x', x'') \geq 0, \forall x \in [0, z]\}.$$

The three sequences  $\{z_s[i]\}$ ,  $s \in \Theta$  in Theorem 6 satisfy

$$z_A[i] = f_A(z_B[i-1], z_C[i-1]),$$

$$z_B[i] = f_B(z_A[i-1], z_C[i-1]),$$

$$z_C[i] = f_C(z_A[i-1], z_B[i-1]).$$

For  $s \in \Theta$ , function  $f_s(\cdot, \cdot)$  is an increasing function for both of its input variables. The following lemma can be proved by the monotonic property of the functions  $f_s$ ,  $s \in \Theta$ .

**Lemma 7.** *The limit  $(z_A^*, z_B^*, z_C^*)$  of the three sequences defined in Theorem 6 is the first intersection of the surfaces  $x = f_A(y, z)$ ,  $y = f_B(x, z)$  and  $z = f_C(x, y)$ ,  $x, y, z \in [0, 1]$ .*

*Proof:* This lemma is a special case of Lemma 14 in Section V. ■

The definition of feasible points can be extended to LC-3 fountain codes. For a point  $(a_A, a_B, a_C)$  in the region  $\{(x_A, x_B, x_C) : 0 \leq x_A, x_B, x_C \leq 1\}$ , we say  $(a_A, a_B, a_C)$  is  $(C_A, C_B, C_C)$ -feasible for an LC-3 fountain code if  $a_A \leq f_A(a_B, a_C; C_A)$ ,  $a_B \leq f_B(a_A, a_C; C_B)$  and  $a_C \leq f_C(a_A, a_B; C_C)$ . The following theorem is useful in deriving the degree-distribution optimization problems for binary LC-3 fountain codes.

**Theorem 8.** *For each  $s \in \Theta = \{A, B, C\}$ , fix  $C_s > R_s > 0$ . Consider a sequence of binary LC-3 fountain codes with  $N = 1, 2, \dots$ , where  $K_s/N \leq R_s$  for  $s \in \Theta$ . For any  $(a_A, a_B, a_C)$ , if there exists a feasible continuous curve  $(x_A(t), x_B(t), x_C(t))$  between the origin and  $(a_A, a_B, a_C)$ , then i) a BP decoding algorithm will stop with at least  $a_s K_s$   $s$ -input packets being decoded for all  $s \in \Theta$  with probability converging to one when  $N \rightarrow \infty$ , and ii) there exists an increasing feasible continuous curve  $(\tilde{x}_A(t), \tilde{x}_B(t), \tilde{x}_C(t))$  between the origin and  $(a_A, a_B, a_C)$ .*

*Proof:* This theorem is a special case of Theorem 15 in Section V. ■

#### D. Optimizations

Fix the parameters defined in Section IV-A. Suppose that we want to design a binary LC-3 fountain codes such that at least  $\eta_s$  fraction of  $s$ -input packets can be decoded by the batched BP decoding for all  $s \in \Theta$ . Theorem 8 converts the problem to the existence of feasible curves: For any triple  $\bar{C} = (C_A, C_B, C_C)$ , if there exists a  $\bar{C}$ -feasible curve between the origin and  $(\eta_A, \eta_B, \eta_C)$ , then the BP decoding will stop with at least  $\eta_s K_s$   $s$ -input packets decoded for all  $s \in \Theta$ , and hence the rate triple  $(\eta_A C_A, \eta_B C_B, \eta_C C_C)$  is *achievable* by the batched BP decoding. Theorem 8 also enables us to consider only the increasing curves from the origin to  $(\eta_A, \eta_B, \eta_C)$ .

By definition, a point  $(\hat{x}_A, \hat{x}_B, \hat{x}_C)$  is  $(C_A, C_B, C_C)$ -feasible if  $\hat{x}_A \leq f_A(\hat{x}_B, \hat{x}_C; C_A)$ ,  $\hat{x}_B \leq f_B(\hat{x}_A, \hat{x}_C; C_B)$  and  $\hat{x}_C \leq f_C(\hat{x}_A, \hat{x}_B; C_C)$ , which are equivalent to

$$F_A(x, \hat{x}_B, \hat{x}_C; C_A) \geq 0, \quad \forall x \in [0, \hat{x}_A],$$

$$F_B(x, \hat{x}_A, \hat{x}_C; C_B) \geq 0, \quad \forall x \in [0, \hat{x}_B],$$

$$F_C(x, \hat{x}_A, \hat{x}_B; C_C) \geq 0, \quad \forall x \in [0, \hat{x}_C],$$

and hence equivalent to

$$\begin{aligned}\Sigma(\Psi_B(\hat{x}_B), \Psi_C(\hat{x}_C))\Psi'_A(x) + C_A/\beta_3 \ln(1-x) &\geq 0, \quad \forall x \in [0, \hat{x}_A], \\ \Sigma(\Psi_A(\hat{x}_A), \Psi_C(\hat{x}_C))\Psi'_B(x) + C_B/\beta_3 \ln(1-x) &\geq 0, \quad \forall x \in [0, \hat{x}_B], \\ \Sigma(\Psi_A(\hat{x}_A), \Psi_B(\hat{x}_B))\Psi'_C(x) + C_C/\beta_3 \ln(1-x) &\geq 0, \quad \forall x \in [0, \hat{x}_C].\end{aligned}$$

We only evaluate the zig-zag type of curves from the origin to  $(\eta_A, \eta_B, \eta_C)$ . Fix a positive integer  $t_{\max}$  and three sequences of real numbers  $0 = x_s[0] \leq x_s[1] \leq \dots \leq x_s[t_{\max}] = \eta_s$ ,  $s \in \Theta$ . The curve formed by line segments

$$(x_A[t], x_B[t], x_C[t]) - (x_A[t+1], x_B[t], x_C[t]) - (x_A[t+1], x_B[t+1], x_C[t]) - (x_A[t+1], x_B[t+1], x_C[t+1])$$

$t = 0, 1, \dots, t_{\max} - 1$  is an increasing zig-zag curve from the origin to  $(\eta_A, \eta_B, \eta_C)$ . Due to the property of the feasible curves, we only need to check the feasibility of the points

$$\begin{aligned}(x_A[0], x_B[0], x_C[0]), (x_A[1], x_B[0], x_C[0]), (x_A[1], x_B[1], x_C[0]), \\ (x_A[1], x_B[1], x_C[1]), (x_A[2], x_B[1], x_C[1]), \dots, (x_A[t_{\max}], x_B[t_{\max}], x_C[t_{\max}]).\end{aligned}\tag{11}$$

We are now ready to introduce the optimization problems for binary LC-3 fountain codes. Fix  $t_{\max}$ ,  $C_B$ ,  $C_C$ ,  $\eta_A$ ,  $\eta_B$  and  $\eta_C$ . The following optimization problem maximizes the achievable rate of source node A for given rates of source nodes B and C:

$$\begin{aligned}\max \quad & \eta_A \theta_A \beta_3 \\ \text{s.t.} \quad & \forall s \in \Theta, x_s[0] = 0, x_s[t_{\max}] = \eta_s; \\ & \forall s \in \Theta, \forall t = 1, \dots, t_{\max}, \quad x_s[t] \geq x_s[t-1]; \\ & \forall t = 1, \dots, t_{\max}, \\ & \Sigma(\Psi_B(x_B[t-1]), \Psi_C(x_C[t-1]))\Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (x_A[t-1], x_A[t]), \\ & \Sigma(\Psi_A(x_A[t]), \Psi_C(x_C[t-1]))\Psi'_B(x) + C_B/\beta_3 \ln(1-x) \geq 0, \quad \forall x \in (x_B[t-1], x_B[t]), \\ & \Sigma(\Psi_A(x_A[t]), \Psi_B(x_B[t]))\Psi'_C(x) + C_C/\beta_3 \ln(1-x) \geq 0, \quad \forall x \in (x_C[t-1], x_C[t]),\end{aligned}\tag{12}$$

where the variables of the optimization are  $\theta_A$ ,  $x_s[t]$ ,  $t = 1, \dots, t_{\max}$ ,  $s \in \Theta$ , degree distributions  $\Psi_A$ ,  $\Psi_B$  and  $\Psi_C$ . The constraints of the above optimization guarantee that the points in (11) are feasible.

Fix  $t_{\max}$ ,  $\eta_A$ ,  $\eta_B$  and  $\eta_C$ . The following optimization problem maximizes the sum rate of the three source nodes:

$$\begin{aligned}\max \quad & \beta_3(\eta_A \theta_A + \eta_B \theta_B + \eta_C \theta_C) \\ \text{s.t.} \quad & \forall s \in \Theta, x_s[0] = 0, x_s[t_{\max}] = \eta_s; \\ & \forall s \in \Theta, \forall t = 1, \dots, t_{\max}, \quad x_s[t] \geq x_s[t-1]; \\ & \forall t = 1, \dots, t_{\max}, \\ & \Sigma(\Psi_B(x_B[t-1]), \Psi_C(x_C[t-1]))\Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (x_A[t-1], x_A[t]), \\ & \Sigma(\Psi_A(x_A[t]), \Psi_C(x_C[t-1]))\Psi'_B(x) + \theta_B \ln(1-x) \geq 0, \quad \forall x \in (x_B[t-1], x_B[t]), \\ & \Sigma(\Psi_A(x_A[t]), \Psi_B(x_B[t]))\Psi'_C(x) + \theta_C \ln(1-x) \geq 0, \quad \forall x \in (x_C[t-1], x_C[t]),\end{aligned}\tag{13}$$

where the variables of the optimization are  $\theta_s, x_s[t], t = 1, \dots, t_{\max}, s \in \Theta$ , degree distributions  $\Psi_A, \Psi_B$  and  $\Psi_C$ .

*Remark 4.* The maximum degree  $D$  can be similarly bounded as in Lemma 4.

*Remark 5.* We can similarly obtain the degree distribution optimization problems for the ordinary BP decoding.

### E. Achievable Rates

Given the distribution  $g$  of the transfer matrix, we know from Section II-B that a rate triple  $(R_A, R_B, R_C)$  is achievable by the binary LC-3 fountain codes only if

$$\begin{aligned}
R_A &\leq \sum_i g(H_i) \text{rk}(H_i^{\{A\}}) = g(H_1) + g(H_4) + g(H_5) + \sum_{i=7}^9 g(H_i) + \sum_{i=11}^{15} g(H_i) \\
&= \beta_3(\alpha_A + \alpha_{A+B} + \alpha_{A+C} + \alpha_{A+B+C} + \bar{\alpha}), \\
R_B &\leq \sum_i g(H_i) \text{rk}(H_i^{\{B\}}) = \beta_3(\alpha_B + \alpha_{A+B} + \alpha_{B+C} + \alpha_{A+B+C} + \bar{\alpha}), \\
R_C &\leq \sum_i g(H_i) \text{rk}(H_i^{\{C\}}) = \beta_3(\alpha_C + \alpha_{B+C} + \alpha_{A+C} + \alpha_{A+B+C} + \bar{\alpha}), \\
R_A + R_B &\leq \sum_i g(H_i) \text{rk}(H_i^{\{A,B\}}) \\
&= g(H_1) + g(H_2) + 2g(H_4) + g(H_5) + g(H_6) + 2g(H_7) \\
&\quad + \sum_{i=8}^{11} g(H_i) + 2g(H_{12}) + 2g(H_{13}) + g(H_{14}) + 2g(H_{15}) \\
&= \beta_3(1 - \alpha_C), \\
R_B + R_C &\leq \sum_i g(H_i) \text{rk}(H_i^{\{B,C\}}) = \beta_3(1 - \alpha_A), \\
R_A + R_C &\leq \sum_i g(H_i) \text{rk}(H_i^{\{A,C\}}) = \beta_3(1 - \alpha_B), \\
R_A + R_B + R_C &\leq \sum_i g(H_i) \text{rk}(H_i) = \beta_3.
\end{aligned}$$

Instead of specifying a value of  $\beta_3$ , we remove  $\beta_3$  from the objective functions of both (12) and (13) so that the optimal values are the normalized (sum) rates. The best numerical results obtained by evaluating (13) are listed in Table II, where we see that the normalized achievable sum rates are all close to 1, the upper bound. One of the vertex of the above region is

$$\begin{aligned}
R_A &= \beta_3(\alpha_A + \alpha_{A+B} + \alpha_{A+C} + \alpha_{A+B+C} + \bar{\alpha}), \\
R_B &= \beta_3(\alpha_B + \alpha_{B+C} + \bar{\alpha}), \\
R_C &= \beta_3\alpha_C.
\end{aligned}$$

We also evaluate (12) with  $C_B/\beta_3 = (\alpha_B + \alpha_{B+C} + \bar{\alpha})/\eta_B$  and  $C_C/\beta_3 = \alpha_C/\eta_C$ .<sup>3</sup> From Table II, readers can verify that the normalized achievable rates of user A are all close to the corresponding values of  $\alpha_A + \alpha_{A+B} + \alpha_{A+C} + \alpha_{A+B+C} + \bar{\alpha}$ .

<sup>3</sup>We need to pick the parameters such that  $(C_B, C_C)$  is an interior point of the projection of the capacity region on the plane  $R_A = 0$ .

TABLE II

ACHIEVABLE RATES OF BINARY LC-3 FOUNTAIN CODES FOR  $\eta_A = \eta_B = \eta_C = 0.98$ . IN BOTH (12) AND (13), THE OBJECTIVE FUNCTIONS ARE MODIFIED BY REMOVING  $\beta_3$ .  $\hat{R}_A/\beta_3$  IS OBTAINED BY SOLVING (12) WITH  $C_B/\beta_3 = (\alpha_B + \alpha_{B+C} + \bar{\alpha})/\eta_B$  AND  $C_C/\beta_3 = \alpha_C/\eta_C$ ,  $\hat{R}_{\text{SUM}}/\beta_3$  IS OBTAINED BY SOLVING (13), AND  $\hat{R}_{\text{SUM}}^o/\beta_3$  IS OBTAINED BY SOLVING A NORMALIZED SUM-RATE MAXIMIZATION PROBLEM FOR THE ORDINARY BP DECODING.

$\alpha_A, \alpha_B, \alpha_C$	$\alpha_{A+B}, \alpha_{A+C}, \alpha_{B+C}$	$\alpha_{A+B+C}$	$\bar{\alpha}$	$\hat{R}_A/\beta_3$	$\hat{R}_{\text{SUM}}^o/\beta_3$	$\hat{R}_{\text{SUM}}/\beta_3$
0.2	0.1	0	0.05	0.4194	0.9592	0.9784
0.2	0	0.1	0.15	0.3957	0.9273	0.9775
0.1	0.1	0.1	0.15	0.4904	0.9099	0.9556
0.1	0	0.1	0.3	0.4521	0.8653	0.9636
0.05	0.05	0	0.35	0.4532	0.8466	0.9628

We also optimize the sum rate of the ordinary BP decoding and give the best rates we obtained in Table II. We see that the batched BP decoding consistently achieves a sum rate above 95% of the optimal value, while the performance of the ordinary BP decoding decreases significantly when  $\bar{\alpha}$  becomes larger. For the normalized rates given in Table II,  $\beta_3$  can be any value in  $(0, 3)$ .

## V. GENERAL LC FOUNTAIN CODES

We now discuss general LC fountain codes for NCMA with  $L$  users, where the base field is not necessarily binary. The coded packets of a fountain code are not required to be generated independently. Specifically, we relax the requirement that the degrees of the coded packets are independent, and assume that the fraction of batches with transfer matrix  $H$  and the degree of the  $s$ -coded packet being  $d_s$  for all  $s \in \Theta$  converges to  $g(H) \prod_{s \in \Theta} \Psi_s[d_s]$  as  $N$  tends to infinity.

### A. Generalized Batched BP Decoders

Both the ordinary BP decoder for LC-2 fountain codes and the batched BP decoder for LC-3 fountain codes can be extended to decode LC- $L$  fountain codes,  $L > 3$ . As discussed, both decoders can perform decoding in rounds with each round having two stages. The first stage is the same for both decoders, while the second stages are different. For general LC- $L$  fountain codes,  $L > 3$ , we have more options to process the coupled output packets in the second stage. We first define a generic (round-based batched BP) decoder of LC- $L$  fountain codes and then discuss several instances of the generic decoder in terms of their different operations in the second stage.

The generic decoder of LC- $L$  fountain codes starts with the first round and each round has two stages:

- Stage 1: The ordinary BP decoding is applied on the  $s$ -output packets to decode the  $s$ -input packets. The decoding in the first stage is equivalent to the decoding of  $L$  LT codes in parallel. The first stage of the first round uses the clean output packets decoded by the physical layer.
- Stage 2: Each batch is processed individually by one of the algorithms to be specified later to recover a number of *clean* output packets for the next round decoding. When no more clean output packets are recovered than the previous round, the decoding stops.

Now we discuss the instances of the generic decoder in terms of the operations in the second stage, where the linear system of equations in (2) is solved. In the following discussion, we fix  $S \subset \Theta$  and assume that in (2), the  $r$ -input packet  $v_r$  has been decoded in the first stage if and only if  $r \in S$ . We describe three instances of the generic decoder.

The first instance of the generic decoder is the extension of the ordinary BP decoder for LC-2 fountain codes, and is called the *BP-substitution decoder*. The  $i$ -th row of  $H$  is also called the  $s$ -th row where  $s$  is the  $i$ -th symbol in  $\Theta$ . Denote by  $H^S$  the submatrix formed by the rows of  $H$  indexed by  $S$ . The second stage of the instance only substitutes the values of  $v_r, r \in S$  into (2) and obtain

$$[v_s, s \in \Theta \setminus S]H^{\Theta \setminus S} = [u_1, \dots, u_B] - [v_r, r \in S]H^S, \quad (14)$$

where the LHS term is known. Since no further operations are applied to process the above linear system, for certain  $s \in \Theta \setminus S$ ,  $v_s$  can be recovered if and only if  $H^{\Theta \setminus S}$  has a column where all the components are zero except for the component at the  $s$ -th row.

Both the second and third instances of the generic decoder can be regarded as the extensions of the batched BP decoder for LC-3 fountain codes. The second instance is called the *BP-BP decoder*, where the (ordinary) BP algorithm is applied in the second stage. The operation in the second stage includes multiple iterations of the following operations (see also Section III-B). The first iteration is the same as the algorithm in the second stage of the BP-substitution decoder. For each of the following iterations, the clean output packets recovered in the last iteration are substituted back into (14) and new clear output packets are found (by searching columns of  $H^{\Theta \setminus S}$  with only one non-zero component). Take (10) as an example. Suppose that  $v_A$  is known. The first iteration of the second stage will recover  $v_C$  and the second iteration of the second stage will recover  $v_B$ .

The third instance is called the *BP-GE decoder*, where Gaussian (Gauss-Jordan) elimination is applied in the second stage. Specifically, in the second stage of the BP-GE decoder, the substitution in the second stage of the BP-substitution decoder is applied first. Following the substitution, Gaussian elimination transforms  $H^{\Theta \setminus S}$  into the reduced column echelon form  $\tilde{H}$ . We then find the clean output packets by searching columns of  $\tilde{H}$  with only one non-zero component. To further reduce the complexity, we can first apply the BP algorithm as in the second stage of the BP-BP decoder and after the BP algorithm stops, apply the Gaussian elimination. Consider the following batch with four users:

$$[u_1, u_2] = [v_A, v_B, v_C, v_D] \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

where  $v_A, \dots, v_D$  are the input packets. Suppose that  $v_A$  is known. The second stage of the BP-BP decoder will stop after the first iteration without any clean output packets recovered. However, the second stage of the BP-GE decoder can recover  $v_B$ .

For the binary LC-2 fountain codes, the BP-substitution, BP-BP and BP-GE decoders are all the same as the



ordinary BP decoder discussed in Section III. For the binary LC-3 fountain codes, the BP-substitution decoder is the same as the ordinary BP decoder discussed in Section IV, and both the BP-BP and BP-substitution decoders are the same as the batched BP decoder discussed in Section IV.

We evaluate the computation complexity of the BP-GE decoder of LC- $L$  fountain codes. The other two instances we discussed have lower complexity. For a batch of  $r$  output packets, the complexity of Gaussian elimination for recovering  $r$  clean output packets is  $O(r^3 + rLT)$  finite-field operations per batch. The total complexity to process all the batches converges to

$$\begin{aligned} & O \left( N \sum_H g(H) [\text{rk}(H)^3 + \text{rk}(H)LT] \right) \\ &= O(N(\beta_L L^2 + \beta_L LT)) \end{aligned} \quad (15)$$

$$= O(\bar{n}(L^2 + LT)), \quad (16)$$

where  $\beta_L$  is defined in (3) and  $\bar{n} = N\beta_L$  is the expected number of output packets. The clean  $s$ -coded packets will be used in the BP decoding of  $s$ -input packets, which has complexity  $O(K_s T)$  finite-field operations. Since  $\bar{n} \geq \sum_s K_s$ , the total decoding complexity is dominated by (16).

If we know that at most  $\tilde{L}$  linear equations can be recovered by NCMA, i.e.,  $\text{rk}(H) \leq \tilde{L}$ , the complexity (15) can be simplified to  $O(\bar{n}(\tilde{L}^2 + LT))$ .

### B. Local Information Function

Instead of analyzing the batched BP decoders defined above individually, we provide a unified analysis of these decoders using the following characterization of different algorithms in the second stage.

Denote by  $\Theta^{\setminus s}$  the set  $\Theta \setminus \{s\}$ . For a set  $S$ , denote by  $2^S$  the collection of all subsets of  $S$ . Recall that  $\mathcal{H}_L$  is the collection of all the full-column-rank,  $L$ -row matrices over  $\mathbb{F}_q$  (see Section II-B). For any  $s \in \Theta$ , the *local information function (LIF)*  $\gamma_s^* : \mathcal{H}_L \rightarrow 2^{\Theta^{\setminus s}}$  is defined by

- 1) for any  $S \in \gamma_s^*(H)$ ,  $v_s$  can be uniquely solved by (2) if the values of  $v_r$ ,  $r \in S$  are all known;
- 2)  $\gamma_s^*(H)$  includes all such subsets of  $\Theta^{\setminus s}$ .

In other words, for any  $S \in \gamma_s^*(H)$ , using linear combinations of the equations in (2), we can obtain the equation

$$v_s = u - \sum_{r \in S} \phi_r v_r, \quad (17)$$

where  $u$  is a linear combination of  $u_1, \dots, u_B$ , and  $\phi_r \in \mathbb{F}_q$ .

Let us illustrate the definition of LIFs by several examples. First consider two special cases. When the row of  $H$  corresponding to  $v_s$  contains only '0's, that is,  $v_s$  is not involved in any output packets of the batch, we have  $\gamma_s^*(H) = \emptyset$ . When in one column of  $H$ , the component corresponding to  $v_s$  is '1' and the rest components are '0's, that is, one of the output packets in the batch is exactly  $v_s$ , we have  $\gamma_s^*(H) = 2^{\Theta^{\setminus s}}$ , i.e., all the subsets of  $\Theta^{\setminus s}$ .

Consider one more example with  $\mathbb{F}_q = \text{GF}(2)$ ,  $\Theta = \{A, B, C, D\}$ , where  $A \leq B \leq C \leq D$ , and the transfer matrix

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (18)$$

We can see that

$$\gamma_A^*(H) = 2^{\{B, C, D\}} \setminus \{\{C\}, \emptyset\},$$

$$\gamma_B^*(H) = 2^{\{A, C, D\}} \setminus \{\{C\}, \emptyset\},$$

$$\gamma_C^*(H) = 2^{\{A, B, D\}},$$

$$\gamma_D^*(H) = 2^{\{A, B, C\}} \setminus \{\{C\}, \emptyset\}.$$

We have the following basic properties of  $\gamma_s^*$ .

**Lemma 9.** *Let  $H$  be an  $L \times B$  full-column rank matrix over  $\mathbb{F}_q$ .*

- 1) *If  $S' \in \gamma_s^*(H)$ , then  $S \in \gamma_s^*(H)$  for any  $S' \subset S \subset \Theta \setminus s$ ;*
- 2)  *$\gamma_s^*(H) = \gamma_s^*(H\Phi)$  for any full-rank  $B \times B$  matrix  $\Phi$ .*

LIFs completely characterize the relations between  $s$ -coded packet and other coded packets in a batch: The  $s$ -coded packet in a batch with transfer matrix  $H$  can be recovered by Gaussian elimination if and only if for certain  $S \in \gamma_s^*(H)$ , all the values of  $v_r, r \in S$  are known. We can also use certain subsets of  $\gamma_s^*(H)$  to characterize the second stages of the BP-substitution and BP-BP decoders.

A function  $\gamma_s : \mathcal{H}_L \rightarrow 2^{\Theta \setminus s}$  is called a *partial LIF* if

- 1)  $\gamma_s(H) \subset \gamma_s^*(H)$ ;
- 2) for any  $S \in \gamma_s(H)$ , all the super sets of  $S$  in  $\Theta \setminus s$  are in  $\gamma_s(H)$ .

For a subset  $\mathcal{A}$  of  $2^{\Theta \setminus s}$ , the span of  $\mathcal{A}$  in  $\Theta \setminus s$ , denoted by  $\langle \mathcal{A} \rangle_{\Theta \setminus s}$ , is the collection of all  $S \subset \Theta \setminus s$  that include at least one element of  $\mathcal{A}$  as a subset.

Let us see an example of partial LIFs. For an  $L \times B$  full-column rank transfer matrix  $H$ , define  $\text{supp}_j(H)$  for  $1 \leq j \leq B$  as the support set of the  $j$ -th column of  $H$ , i.e., the subset of  $s \in \Theta$  such that the component of  $H$  on the  $s$ -th row,  $j$ -th column is nonzero. For the  $H$  in (18), we have

$$\text{supp}_1(H) = \{A, D\},$$

$$\text{supp}_2(H) = \{B, D\},$$

$$\text{supp}_3(H) = \{C\}.$$

Define

$$\gamma_s^o(H) = \langle \{\text{supp}_i(H) \setminus \{s\}, i \in \{1, \dots, B\} : s \in \text{supp}_i(H)\} \rangle_{\Theta \setminus s}.$$

We see that  $\gamma_s^o$  is a partial LIF since if  $s \in \text{supp}_i(H)$  then  $\text{supp}_i(H) \setminus \{s\} \in \gamma_s^*(H)$ . For the  $H$  in (18), we have

$$\begin{aligned}\gamma_A^o(H) &= \langle \{\{D\}\} \rangle_{\{B,C,D\}}, \\ \gamma_B^o(H) &= \langle \{\{D\}\} \rangle_{\{A,C,D\}}, \\ \gamma_C^o(H) &= \langle \emptyset \rangle_{\{A,B,D\}}, \\ \gamma_D^o(H) &= \langle \{\{A\}, \{B\}\} \rangle_{\{A,B,C\}}.\end{aligned}$$

For a given linear system (14) and  $s \in \Theta$ ,  $\gamma_s^o(H)$  gives all the possible ways to solve  $v_s$  without any matrix operations. Therefore,  $\gamma_s^o$  characterizes the second stage of the BP-substitution decoder.

Let us continue to discuss how to characterize the second stage of the BP-BP decoder. From the above discussion,  $\gamma_s^o$  tells us the solvability of  $v_s$  using one iteration of the BP algorithm on (14). Define  $\gamma_s^{b,1} = \gamma_s^o$ . For  $i = 2, \dots, L$ , define function  $\gamma_s^{b,i} : \mathcal{H}_L \rightarrow 2^{\Theta \setminus s}$  as

$$\gamma_s^{b,i}(H) = \gamma_s^{b,i-1}(H) \cup \left( \bigcup_{T \in \gamma_s^o(H)} \tilde{\gamma}_s^{i-1}(T, H) \right),$$

where

$$\tilde{\gamma}_s^{i-1}(T, H) = \left\{ \bigcup_{r \in T} T_r : s \notin T_r \in \gamma_r^{b,i-1}(H), \forall r \in T \right\}.$$

The following lemma tells that  $\{\gamma_s^{b,i}, s \in \Theta\}$  characterizes the first  $i$  iterations of the second stage of the BP-BP decoder.

**Lemma 10.** *For  $i = 1, \dots, L$ ,  $\gamma_s^{b,i}$  are partial LIFs; and for  $S \in \gamma_s^{b,i}(H)$ ,  $v_s$  can be solved in terms of  $v_r, r \in S$  using at most  $i$  iterations of the ordinary BP algorithm on the linear system (2).*

*Proof:* We prove the lemma by induction. First the above claims hold for  $i = 1$ . Fix  $i > 1$ . For any  $S \in \gamma_s^{b,i}(H)$ , either  $S \in \gamma_s^{b,i-1}(H)$  or  $S \in \tilde{\gamma}_s^{i-1}(T, H)$  for certain  $T \in \gamma_s^o(H)$ . If  $S \in \gamma_s^{b,i-1}(H)$ , by the induction hypothesis,  $v_s$  can be solved using at most  $i - 1$  iterations of the ordinary BP algorithm, and all the supersets of  $S$  in  $\Theta \setminus s$  are in  $\gamma_s^{b,i-1}(H)$  and hence in  $\gamma_s^{b,i}(H)$ . If  $S \in \tilde{\gamma}_s^{i-1}(T, H)$ , then  $S = \bigcup_{r \in T} T_r$  for certain  $T_r \in \gamma_r^{b,i-1}(H), s \notin T_r$ . By induction hypothesis,  $v_r$  can be solved using at most  $i - 1$  iterations of the ordinary BP algorithm in terms of  $v_{r'}, r' \in T_r$ . Since  $T \in \gamma_s^o(H)$ , we can use one more iteration of the BP algorithm to recover  $v_s$  in terms of  $v_{r'}, r' \in \bigcup_{r \in T} T_r$ . Further, for any  $S' \supset S, S' \subset \Theta \setminus s$ , we can write  $S' = \bigcup_{r \in T} T'_r$ , where  $T'_r \supset T_r, r \in T$ . Since  $s \notin T'_r \in \gamma_r^{b,i-1}(H)$ , we have  $S' \in \tilde{\gamma}_s^{i-1}(T, H)$ . This completes the proof of the lemma.  $\blacksquare$

We say a batched BP decoder of LC- $L$  fountain codes is characterized by partial LIFs  $\{\gamma_s, s \in \Theta\}$  if the second stage of each round of the BP decoder satisfies the following property: For each batch with transfer matrix  $H$ , known values of  $v_r, r \in S$  and any  $s \in \Theta \setminus S$ ,  $v_s$  can be recovered if and only if  $S \in \gamma_s(H)$ . Specifically, the BP-substitution decoder, the BP-BP decoder with  $i$  iterations in the second stage, and the BP-GE decoder are the batched BP decoders characterized by  $\{\gamma_s^o, s \in \Theta\}$ ,  $\{\gamma_s^{b,i}, s \in \Theta\}$ , and  $\{\gamma_s^*, s \in \Theta\}$ , respectively. We will analyze a general batched BP decoder characterized by any partial LIFs  $\{\gamma_s, s \in \Theta\}$ .

### C. Analysis of Decoding

We analyze the performance of the batched BP decoder characterized by partial LIFs  $\{\gamma_s, s \in \Theta\}$ . For  $s \in \Theta$ , transfer matrix  $H$  and  $0 \leq y_r \leq 1, r \in \Theta \setminus s$ , define

$$\Gamma_s(H, y_r, r \in \Theta \setminus s) = \sum_{S \in \gamma_s(H)} \prod_{r \in S} y_r \prod_{r \in \Theta \setminus (\{s\} \cup S)} (1 - y_r). \quad (19)$$

Suppose that a batch is generated by  $\{v_s, s \in \Theta\}$ . If with probability  $p_r$ , the value of  $v_r$  is known, then the probability that  $v_s$  can be expressed as the already-known  $v_r, r \in \Theta \setminus s$  by the relations given in  $\gamma_s(H)$  is exactly  $\Gamma_s(H, p_r, r \in \Theta \setminus s)$ . For example, when  $\gamma_s(H) = \emptyset$ , the value of  $\Gamma_s(H, p_r, r \in \Theta \setminus s)$  is zero; when  $\gamma_s(H) = 2^{\Theta \setminus s}$ , the value of  $\Gamma_s(H, p_r, r \in \Theta \setminus s)$  is one.

**Theorem 11.** *For each  $s \in \Theta$ , fix  $C_s > R_s > 0$ . Consider an LC-L fountain codes with  $N$  batches employing a batched BP decoder characterized by partial LIFs  $\{\gamma_s, s \in \Theta\}$ , where  $K_s/N \leq R_s$  for  $s \in \Theta$ . Define for  $s \in \Theta$*

$$F_s(x, y_r, r \in \Theta \setminus s) = F_s(x, y_r, r \in \Theta \setminus s; C_s) = \Psi'_s(x) + \frac{C_s}{\sum_H g(H) \Gamma_s(H, \Psi_r(y_r), r \in \Theta \setminus s)} \ln(1 - x).$$

*Let  $z_s[0] = 0$  and for  $i \geq 1$  let  $z_s[i]$  be the maximum value of  $z$  such that for any  $x \in [0, z]$ , we have*

$$F_s(x, z_r[i-1], r \in \Theta \setminus s) \geq 0.$$

*The sequence  $\{z_s[i]\}$  is increasing and upper bounded. Let  $z_s^*$  be the limit of the sequence  $\{z_s[i]\}$ . Then there exists a positive number  $c$  such that when  $N$  is sufficiently large, with probability at least  $1 - e^{-cN}$ , the batched BP decoder stops with at least  $z_s^* K_s$   $s$ -input packets being decoded for all  $s \in \Theta$ .*

*Remark 6.* Since  $\gamma_s^o(H) \subseteq \gamma_s^*(H)$  for all  $s \in \Theta$ , the value of  $\Gamma_s$  with respect to  $\gamma_s^*(H)$  is larger than or equal to the value of  $\Gamma_s$  with respect to  $\gamma_s^o(H)$ . Therefore, in general the performance of batched BP decoding is better than the performance of ordinary BP decoding.

The proof of the above theorem is postponed to the next subsection. Let us show how to apply the above theorem to the binary LC-2 and LC-3 fountain codes. The binary LC-2 fountain code has four non-trivial transfer matrices (see (4)). The batched BP decoder reduces to the ordinary BP decoder, i.e.,  $\gamma_s^*(H_i) = \gamma_s^o(H_i), i = 1, \dots, 4$ . We can calculate that for  $\gamma_s = \gamma_s^*$ ,

$$\begin{aligned} \sum_i g(H_i) \Gamma_A(H_i, y_B) &= g(H_1) + g(H_3)y_B + g(H_4), \\ \sum_i g(H_i) \Gamma_B(H_i, y_A) &= g(H_2) + g(H_3)y_A + g(H_4). \end{aligned}$$

Recall that  $\beta_2 = g(H_1) + g(H_2) + g(H_3) + 2g(H_4)$ . The proof of Theorem 1 is completed by substituting  $\alpha_A = \frac{g(H_1)+g(H_4)}{\beta_2}$ ,  $\alpha_B = \frac{g(H_2)+g(H_4)}{\beta_2}$  and  $\alpha_{A+B} = \frac{g(H_3)}{\beta_2}$  into Theorem 11.

The binary LC-3 fountain code has 17 non-trivial transfer matrices (see Section IV-A). The batched BP decoder of the binary LC-3 fountain code is characterized by  $\{\gamma_s^*, s \in \{A, B, C\}\}$ . Recall the parameters defined in

Section IV-A. We can calculate that when  $\gamma_s = \gamma_s^*$ ,

$$\begin{aligned}\sum_{i=1}^{17} g(H_i) \Gamma_A(H_i, y_B, y_C) / \beta_3 &= \alpha_A + \alpha_{A+B} y_B + \alpha_{A+C} y_C + \alpha_{A+B+C} y_B y_C + \bar{\alpha}(y_B + y_C - y_B y_C), \\ \sum_{i=1}^{17} g(H_i) \Gamma_B(H_i, y_A, y_C) / \beta_3 &= \alpha_B + \alpha_{A+B} y_A + \alpha_{B+C} y_C + \alpha_{A+B+C} y_A y_C + \bar{\alpha}(y_A + y_C - y_A y_C), \\ \sum_{i=1}^{17} g(H_i) \Gamma_C(H_i, y_A, y_B) / \beta_3 &= \alpha_C + \alpha_{A+C} y_A + \alpha_{B+C} y_B + \alpha_{A+B+C} y_A y_B + \bar{\alpha}(y_A + y_B - y_A y_B).\end{aligned}$$

The proof of Theorem 6 is completed by substituting the above three equalities into Theorem 11.

We now apply Theorem 11 to the binary LC-3 fountain code with the ordinary BP decoding, which is characterized by  $\{\gamma_s^o, s \in \{A, B, C\}\}$ . We can calculate that when  $\gamma_s = \gamma_s^o$ ,

$$\begin{aligned}\sum_{i=1}^{17} g(H_i) \Gamma_A(H_i, y_B, y_C) / \beta_3 &= \alpha_A + \alpha_{A+B} y_B + \alpha_{A+C} y_C + \alpha_{A+B+C} y_B y_C + \bar{\alpha}_A(y_B + y_C - y_B y_C) + \bar{\alpha}_B y_B + \bar{\alpha}_C y_C, \\ \sum_{i=1}^{17} g(H_i) \Gamma_B(H_i, y_A, y_C) / \beta_3 &= \alpha_B + \alpha_{A+B} y_A + \alpha_{B+C} y_C + \alpha_{A+B+C} y_A y_C + \bar{\alpha}_A y_A + \bar{\alpha}_B(y_A + y_C - y_A y_C) + \bar{\alpha}_C y_C, \\ \sum_{i=1}^{17} g(H_i) \Gamma_C(H_i, y_A, y_B) / \beta_3 &= \alpha_C + \alpha_{A+C} y_A + \alpha_{B+C} y_B + \alpha_{A+B+C} y_A y_B + \bar{\alpha}_A y_A + \bar{\alpha}_B y_B + \bar{\alpha}_C(y_A + y_B - y_A y_B).\end{aligned}$$

The proof of Theorem 5 is completed by substituting the above three equalities into Theorem 11.

#### D. Proof of Theorem 11

The proof of Theorem 11 uses an existing result for LT codes. The following proposition is implied by [26] and can be proved using the AND-OR tree approach [25].

**Proposition 12.** Fix  $0 < R < C \leq 1$ . Consider an LT code with  $K$  input packets and  $n \geq K/R$  coded packets, where the empirical degree distribution of the coded packets converges in probability to a degree distribution  $\Psi$  with a fixed maximum degree. For any  $0 < \eta < 1$ , if

$$\Psi'(x) + C \ln(1 - x) \geq 0, \forall x \in [0, \eta], \quad (20)$$

then there exists a positive number  $c$  such that when  $n$  is sufficiently large, with probability at least  $1 - \exp(-cn)$ , the BP decoder is able to recover at least  $\eta K$  input packets.

*Proof of Theorem 11:* In the analysis, we introduce an extra criterion to stop the first stage of each round: If the first stage does not stop after  $K_s z_s[i]$   $s$ -input packets have been decoded, we force the first stage to stop. For  $s \in \Theta$ , define random variable  $K_s[i]$  as the total number of decoded  $s$ -input packets after the  $i$ th round. We always have  $K_s[i] \leq K_s z_s[i]$ . We prove by induction that for a sufficiently large  $N$  and  $i = 1, 2, \dots$ ,

$$\Pr \{K_s[i] = K_s z_s[i], s \in \Theta\} = 1 - O(i \exp(-cN)). \quad (21)$$

For a batch transfer matrix  $H$ , let  $\Omega_H$  be the set of all batches with transfer matrix  $H$ . Define

$$\delta_0 = 1 - \max_s (R_s / C_s)^{1/(L+1)}.$$

Henceforth in the proof, we assume that

$$|\Omega_H| \geq Ng(H)(1 - \delta_0), \text{ for all transfer matrices } H \quad (22)$$

holds. Since  $|\Omega_H|/N$  converges to  $g(H)$  for all transfer matrix  $H$ , this assumption holds for sufficiently large  $N$ .

We first prove (21) for  $i = 1$ . Consider the first strage of the first round. Define

$$U_H^0(s) = \begin{cases} \Omega_H & \text{if } \emptyset \in \gamma_s(H), \\ \emptyset & \text{otherwise.} \end{cases}$$

We know that when  $\emptyset \in \gamma_s(H)$ , all  $s$ -coded packets embedded in the batches in  $\Omega_H$  can be recovered and hence can be used in the BP decoding at the first round. Let

$$U^0(s) = \cup_H U_H^0(s)$$

be the batches that can be used in the BP decoding of the  $s$ -input packets at the first round. For  $s \in \Theta$  such that  $|U^0(s)| = 0$ , we have  $K_s[1] = 0$ . Since  $\emptyset \notin \gamma_s(H)$  for all  $H$  in this case, we have  $\sum_H g(H)\Gamma_s(H, 0, \dots, 0) = 0$  and hence  $z_s[1] = 0$  according to the definition in theorem. Therefore,  $K_s[1] = K_s z_s[1]$ . Fix an  $s \in \Theta$  such that  $|U^0(s)| > 0$ . Since the empirical degree distribution of the  $s$ -coded packets embedded in batches in  $U_H^0(s)$  converges to  $\Psi_s$  when  $U_H^0(s) \neq \emptyset$ , we can apply Proposition 12 on the ordinary BP decoding of the  $s$ -coded packets embedded in the batches in  $U^0(s)$ . By (22), we have

$$|U^0(s)| \geq N \sum_H g(H)\Gamma_s(H, 0, \dots, 0)(1 - \delta_0),$$

which implies

$$\frac{K_s}{|U^0(s)|} \leq \frac{R_s}{\sum_H g(H)\Gamma_s(H, 0, \dots, 0)(1 - \delta_0)} < \frac{C_s}{\sum_H g(H)\Gamma_s(H, 0, \dots, 0)}.$$

By the definition of  $z_s[1]$  in the theorem, we see that (20) holds with  $z_s[1]$ ,  $\Psi_s$  and  $\frac{C_s}{\sum_H g(H)\Gamma_s(H, 0, \dots, 0)}$  in place of  $\eta$ ,  $\Psi$  and  $C$ , respectively, and hence (21) with  $i = 1$  is proved by Proposition 12 and the union bound.

Assume that (21) holds for certain  $i \geq 1$ . Suppose that after the first stage of the  $i$ -th round,

$$K_s[i] = K_s z_s[i], \forall s \in \Theta, \quad (23)$$

which holds with probability at least  $1 - O(i \exp(-cN))$  by the induction hypothesis. Suppose that the set  $U_H^{i-1}(s)$  has been assigned, and only the batches in  $U^{i-1}(s) := \cup_H U_H^{i-1}(s)$  are used in the decoding of the  $s$ -input packets at the first stage of the  $i$ -th round.

Consider the second stage of the  $i$ -th round. For a batch  $b$ , denote by  $v_s(b)$  the  $s$ -coded packet embedded in the batch. We say that  $v_s(b)$  is *BP decodable after  $i$ -th rounds* if  $v_s(b)$  is the linear combination of the decoded  $s$ -input packets in the first stage of the  $i$ -th rounds. Denote by  $p_s[i]$  the probability that for a randomly selected batch  $b \notin U^{i-1}(s)$ ,  $v_s(b)$  is BP decodable after the  $i$ -th round of decoding. Since the neighbors of a coded packets are chosen uniformly at random, conditioning on the event in (23), we have

$$p_s[i] \geq \sum_d \Psi_s[d](1 - \delta_0/2) \frac{\binom{K_s z_s[i]}{d}}{\binom{K_s}{d}} \geq \Psi_s(z_s[i])(1 - \delta_0), \quad (24)$$

where the inequalities hold for sufficiently large  $K_s$ . Let  $\delta_1 = \min_s(1/\Psi_s(z_s^*) - 1)\delta_0$ . On the other hand, we have for sufficiently large  $K_s$ ,

$$p_s[i] \leq \sum_d \Psi_s[d](1 + \delta_1) \frac{\binom{K_s z_s[i]}{d}}{\binom{K_s}{d}} \leq \Psi_s(z_s[i])(1 + \delta_1),$$

which implies

$$1 - p_s[i] \geq 1 - \Psi_s(z_s[i])(1 + \delta_1) \geq (1 - \Psi_s(z_s[i]))(1 - \delta_0). \quad (25)$$

For a set  $U$ , let  $\text{Sa}(U, p)$  be a subset of  $U$  where each element in  $U$  is chosen with probability  $p$  independently. Define

$$\begin{aligned} D_H^i(s) &= \{b \in \Omega_H \setminus U_H^{i-1}(s) : v_s(b) \text{ is BP decodable after } i\text{-th rounds}\} \cup \text{Sa}(U_H^{i-1}(s), p_s[i]), \\ D_H^i(s, S) &= \cap_{r \in S} D_H^i(r) \setminus \cup_{r' \notin \{s\} \cup S} D_H^i(r'), \quad S \subset \Theta \setminus s, \\ U_H^i(s) &= \cup_{S \in \gamma_s(H)} D_H^i(s, S). \end{aligned}$$

For a batch  $b \in D_H^i(r)$ , the  $r$ -coded packet embedded in  $b$  is either BP decodable after  $i$  rounds (when  $b \notin U_H^{i-1}(r)$ ) or known before the  $i$ -th rounds (when  $b \in U_H^{i-1}(r)$ ). So for  $s \notin S \subset \Theta$  and batch  $b \in D_H^i(s, S)$ , all  $v_r(b), r \in S$  are known after the first stage of the  $i$ -th round. If we further have  $S \in \gamma_s(H)$ ,  $v_s(b)$  can be recovered in terms of  $v_r, r \in S$ . Therefore, for all the batches  $b$  in  $U_H^i(s)$ , the  $s$ -coded packets embedded in  $b$  can be recovered at the second stage of the  $i$ -th round, and hence can be used in the BP decoding of the  $(i+1)$ -th round.

Turn to the first stage of the  $(i+1)$ -th round. Let  $U^i(s) = \cup_H U_H^i(s)$ . To apply Proposition 12 on the ordinary BP decoding of the  $s$ -input packet at the  $(i+1)$ -th round, we need to verify the degree distribution of the  $s$ -coded packets recovered from the batches in  $U^i(s)$  and count the cardinality of  $U^i(s)$ . Each batch  $b \in \Omega_H$  is in  $D_H^i(s, S)$  independently with probability  $\prod_{r \in S} p_r[i] \prod_{r' \notin \{s\} \cup S} (1 - p_{r'}[i])$ . So the degree distribution of the  $s$ -coded packets embedded in the batches in  $D_H^i(s, S)$  converges in probability to  $\Psi_s$  as  $N$  tends to infinity. Since  $D_H^i(s, S), S \subset \Theta \setminus s$  form a partition of  $\Omega_H$ , we have

$$|U^i(s)| = \sum_H \sum_{S \in \gamma_s(H)} |D_H^i(s, S)|,$$

and hence

$$\mathbb{E}[|U^i(s)|] = \sum_H |\Omega_H| \Gamma_s(H, p_r[i], r \in \Theta \setminus s).$$

Define event  $E_N^i$  as

$$|U^i(s)| \geq \sum_H |\Omega_H| \Gamma_s(H, p_r[i], r \in \Theta \setminus s)(1 - \delta_0), \quad \forall s \in \Theta.$$

By the Chernoff bound, event  $E_N^i$  holds with probability at least  $1 - O(\exp(-c(\delta_0)N))$ , where  $c(\delta_0) > 0$  is a function of  $\delta_0$ . Under the condition that the event  $E_N^i$  holds, together with (22), (24) and (25), we have

$$|U^i(s)| \geq N \sum_H g(H) \Gamma_s(H, \Psi_r(z_r[i]), r \in \Theta \setminus s)(1 - \delta_0)^{L+1},$$

which implies

$$\frac{K_s}{|U^i(s)|} \leq \frac{R_s}{\sum_H g(H) \Gamma_s(H, \Psi_r(z_r[i]), r \in \Theta^{\setminus s})(1 - \delta_0)^{L+1}} < \frac{C_s}{\sum_H g(H) \Gamma_s(H, \Psi_r(z_r[i]), r \in \Theta^{\setminus s})}.$$

By the definition of  $z_s[i+1]$  in the theorem, we see that (20) holds with  $z_s[i+1]$ ,  $\Psi_s$  and  $\frac{C_s}{\sum_H g(H) \Gamma_s(H, \Psi_r(z_r[i]), r \in \Theta^{\setminus s})}$  in place of  $\eta$ ,  $\Psi$  and  $C$ , respectively. By Proposition 12, when  $N$  is sufficiently large, we have

$$\Pr \{K_s[i+1] \geq K z_s[i+1] | K_r[i] = K_r z_r[i], \forall r \in \Theta\} = 1 - O(\exp(-cN)),$$

where the probability that event  $E_N^i$  holds is counted by modifying  $c$ . Using the union bound and counting the probability that (23) holds, (21) is proved with  $i+1$  in place of  $i$ .

We only need to run at most  $\sum_s K_s$  rounds of the decoding algorithm before no new input packets can be decoded. Therefore, with probability  $1 - O(N \exp(-cN))$ , a BP decoding algorithm stops with at least  $z_s K_s$   $s$ -variable decoded for all  $s \in \Theta$ . The proof is completed by decreasing  $c$  slightly. ■

### E. Geometric Characterization

For  $s \in \Theta$ , define

$$f_s(y_r, r \in \Theta^{\setminus s}) = f_s(y_r, r \in \Theta^{\setminus s}; C_s) = \max \left\{ z : F_s(x, y_r, r \in \Theta^{\setminus s}; C_s) \geq 0, \forall x \in [0, z] \right\}.$$

The sequences  $\{z_s[i]\}$ ,  $s \in \Theta$  defined in Theorem 11 satisfy

$$z_s[i] = f_s(z_r[i-1], r \in \Theta^{\setminus s}).$$

With the help of the following lemma, we see that  $f_s$  is an increasing function for all the input variables.

**Lemma 13.** *For any  $t \in \Theta^{\setminus s}$ ,  $\Gamma_s(H, p_r, r \in \Theta^{\setminus s})$  is an increasing function of  $p_t$  with any given values of  $p_r \in [0, 1]$ ,  $r \in \Theta \setminus \{s, t\}$ .*

*Proof:* First, for all  $S \in \gamma_s(H)$  with  $t \in S$ , the derivative of  $\prod_{r \in S} p_r \prod_{r' \notin \{s\} \cup S} (1 - p_{r'})$  for  $p_t$  is nonnegative. Suppose that  $S \in \gamma_s(H)$ ,  $t \notin S$ . Since  $S \cup \{t\} \in \gamma_s(H)$ , by definition  $\Gamma_s(H, p_r, r \in \Theta^{\setminus s})$  includes the summation of two terms:

$$\begin{aligned} & \prod_{r \in S} p_r \prod_{r' \notin S \cup \{s\}} (1 - p_{r'}), \\ & \prod_{r \in S \cup \{t\}} p_r \prod_{r' \notin S \cup \{s, t\}} (1 - p_{r'}). \end{aligned}$$

The derivatives of these two terms for  $p_t$  are

$$\begin{aligned} & - \prod_{r \in S} p_r \prod_{r' \notin S \cup \{s, t\}} (1 - p_{r'}), \\ & \prod_{r \in S} p_r \prod_{r' \notin S \cup \{s, t\}} (1 - p_{r'}), \end{aligned}$$

respectively. Since the summation of these two derivatives is zero, the derivative of  $\Gamma_s(H, p_r, r \in \Theta^{\setminus s})$  for  $p_t$  is nonnegative. ■



The following lemma gives a geometric characterization of the limits of the sequences  $\{z_s[i]\}$ ,  $s \in \Theta$  defined in Theorem 11.

**Lemma 14.** *The point  $(z_s^*, s \in \Theta)$  of the limits of the sequences defined in Theorem 11 is an intersection of the surfaces  $y_s = f_s(y_r, r \in \Theta \setminus s)$ ,  $s \in \Theta$ , and for any point  $(x_r^*, r \in \Theta)$  on the intersection of  $y_s = f_s(y_r, r \in \Theta \setminus s)$ ,  $s \in \Theta$ ,  $z_s^* \leq x_s^*$  for all  $s \in \Theta$ . In other words,  $(z_s^*, s \in \Theta)$  is the first intersection of the surfaces  $y_s = f_s(y_r, r \in \Theta \setminus s)$ ,  $s \in \Theta$ .*

*Proof:* The lemma can be proved using the monotonic property of functions  $f_s$ . Since  $(z_{s'}[i], s' < s, z_s[i+1], z_{s''}[i], s'' > s)$  is on  $y_s = f_s(y_r, r \in \Theta \setminus s)$  for all  $s \in \Theta$ , the limit point  $(z_s^*, s \in \Theta)$  is on  $y_s = f_s(y_r, r \in \Theta \setminus s)$  for all  $s \in \Theta$ . The existence of the intersections of  $y_s = f_s(y_r, r \in \Theta \setminus s)$  for all  $s \in \Theta$  is guaranteed by the existence of the limits of the sequences  $\{z_s[i]\}$ ,  $s \in \Theta$ .

Let  $(x_r^*, r \in \Theta)$  be an intersection of  $y_s = f_s(y_r, r \in \Theta \setminus s)$  for all  $s \in \Theta$ . We show that  $z_s[i] \leq x_s^*$ ,  $s \in \Theta$  by induction. First, by definition  $z_s[0] = 0 \leq x_s^*$ ,  $s \in \Theta$ . Assume that  $z_s[j] \leq x_s^*$ ,  $s \in \Theta$  for some  $j \geq 0$ . Since  $f_s$  is an increasing function of all the input variables, we have  $z_s[j+1] = f_s(z_r[j], r \in \Theta \setminus s) \leq f_s(x_r^*, r \in \Theta \setminus s) = x_s^*$ . Therefore,  $z_s^* \leq x_s^*$  for all  $s \in \Theta$ , and hence the first intersection is well defined. ■

Let  $\mathbf{C} = (C_s, s \in \Theta)$ . We say a point  $(a_r, r \in \Theta)$  in the region  $\{(x_r, r \in \Theta) : 0 \leq x_r \leq 1\}$  is **C-feasible** for an LC-L fountain code if  $a_s \leq f_s(a_r, r \in \Theta \setminus s; C_s)$ . A curve is **C-feasible** for an LC-L fountain code if every point on the curve is **C-feasible**. A point/curve is said to be *feasible* when **C** is implied. One property of feasible points is that if both  $(a_r, r \in \Theta)$  and  $(b_r, r \in \Theta)$  are **C-feasible**, where  $a_s > b_s$  and  $a_r = b_r$  for  $r \in \Theta \setminus s$ , then the segment between these two points is **C-feasible**. The reason is that for any  $x \in (b_s, a_s)$ , we have  $x \leq a_s \leq f_s(a_r, r \in \Theta \setminus s)$  and for  $r \in \Theta \setminus s$ ,  $a_r = b_r \leq f_r(b_t, t \in \Theta \setminus r) \leq f_r(b_t, t \neq r < s, x, b_{t'}, t' \neq r > s)$  (since  $f_r$  is an increasing function for all input variables).

**Theorem 15.** *For each  $s \in \Theta$ , fix  $C_s > R_s > 0$ . Consider an LC-L fountain codes with  $N$  batches employing a batched BP decoder characterized by partial LIFs  $\{\gamma_s, s \in \Theta\}$ , where  $K_s/N \leq R_s$  for  $s \in \Theta$ . For any point  $(a_r, r \in \Theta)$ , if there exists a **C-feasible** continuous curve  $(x_r(t), r \in \Theta)$  between the origin and  $(a_r, r \in \Theta)$ , then i) the batched BP decoder will stop with at least  $a_s K_s$   $s$ -input packets decoded for all  $s \in \Theta$  with probability at least  $1 - e^{-cN}$  when  $N$  is sufficiently large, where  $c$  is a constant value, and ii) there exists an increasing **C-feasible** continuous curve  $(\tilde{x}_r(t), r \in \Theta)$  between the origin and  $(a_r, r \in \Theta)$ .*

*Proof:* Suppose there exists a feasible continuous curve  $V(t) = (x_r(t), r \in \Theta)$  between the origin and  $(a_r, r \in \Theta)$ . We first prove ii) by constructing an increasing feasible continuous curve  $(\tilde{x}_r(t), r \in \Theta)$ . For a given  $s \in \Theta$ , we will show in the next paragraph that we can modify  $V(t)$  to a feasible continuous curve  $V_s(t) = (x'_r(t), r \in \Theta)$  between the origin and  $(a_r, r \in \Theta)$  where  $x'_s(t)$  is an increasing function of  $t$  and for  $r \neq s$ ,  $x'_r(t) = x_r(t)$ . Then we can apply the above modification to all the coordinations successively to obtain an increasing feasible continuous curve  $(\tilde{x}_r(t), r \in \Theta)$  between the origin and  $(a_r, r \in \Theta)$ .

Find the smallest  $t'$  such that  $x_s(t') = a_s$ . We modify  $(x_r(t), r \in \Theta)$  by replacing the part after  $t = t'$  with a line

segment between  $(x_r(t'), r \in \Theta)$  and  $(a_r, r \in \Theta)$  without changing  $x_r(t)$ ,  $t \geq t'$  for all  $r \neq s$ . The new curve is still feasible and continuous and has the same parametric coordinate functions for all the positions other than  $s$ . We use the same notation for the coordination function at the position of  $s$ . The curve  $V_s(t)$  is then formed as follows: start from  $t = 0$ ,  $V_s(t)$  is the same as  $V(t)$  until  $t$  increases to  $\tau$  such that  $x_s(\tau)$  is a local maximum point of  $x_s(t)$ . Find  $\tau'$  as the largest  $t \geq \tau$  such that  $x(t) = x(\tau)$ . We extend  $V_s(t)$  from  $(x_r(\tau), r \in \Theta)$  to  $(x_r(\tau'), r \in \Theta)$  by a line segment without changing  $x_r(t)$ ,  $\tau \leq t \leq \tau'$  for all  $r \neq s$ . Repeat the above procedure from  $t = \tau'$  until the end of the curve is reached. We see that  $x'_s(t)$  is increasing and ends at  $a_s$ , and for  $r \neq s$ ,  $x'_r(t) = x_r(t)$ . This completes the proof of ii).

We prove i) by assuming that curve  $V(t)$  is increasing. Fix any  $\tilde{C}'_r$  such that  $\tilde{R}_r < \tilde{C}'_r < \tilde{C}_r$  for all  $r \in \Theta$ . Let  $(b_r, r \in \Theta)$  be any intersection of  $y_s = f_s(y_r, r \in \Theta \setminus s; \tilde{C}')$ ,  $s \in \Theta$ . If  $b_r \geq a_r$  for all  $r \in \Theta$ , the claim of the theorem holds by Lemma 14 and Theorem 11. In the following, we show by contradiction that it is not possible that  $b_r < a_r$  for certain  $r \in \Theta$ . Without loss of generality, suppose that for certain  $s \in \Theta$ ,  $b_r < a_r$  for all  $r \leq s$  and  $b_r \geq a_r$  for all  $r > s$ . Since the curve  $V(t)$  is increasing, continuous and ends at  $(a_r, r \in \Theta)$ , it must cross a point  $(c_r, r \in \Theta)$  satisfying

- 1)  $c_t = b_t < a_t$  for certain  $t \leq s$ ,
- 2)  $c_r \leq b_r < a_r$  for  $r \neq t \leq s$ , and
- 3)  $c_r \leq a_r \leq b_r$  for all  $r > s$ .

We have

$$\begin{aligned} c_t &= b_t \\ &= f_t(b_r, r \neq t \leq s, b_{r'}, r' > s; C') \end{aligned} \tag{26}$$

$$\geq f_t(c_r, r \neq t \leq s, c_{r'}, r' > s; C') \tag{27}$$

$$> f_t(c_r, r \neq t \leq s, c_{r'}, r' > s; C), \tag{28}$$

where (26) follows that  $(b_r, r \in \Theta)$  is on  $y_t = f_t(\cdots; C')$ ; and (27) and (28) are obtained using the monotonic property of  $f_t$ . Since (28) implies that  $(c_r, r \in \Theta)$  is not feasible, we obtain a contradiction to that  $(c_r, r \in \Theta)$  is on  $V(t)$ . Therefore,  $a_r \leq b_r$  for all  $r$  and the proof is completed.  $\blacksquare$

## VI. CONCLUDING REMARKS

Motivated by NCMA, we analyzed and designed near optimal linearly-coupled fountain codes for linear multiple-access channels. The coupling of codes is a general phenomenon when network coding is used in a network with multiple source nodes. To the best of our knowledge, our work provides the first analysis of the joint BP decoding of messages from multiple sources coupled by network coding. Leveraging on the simplicity of batched BP decoding, our framework may find application in many practical multi-source communication systems besides NCMA.

## APPENDIX

### SOLVING THE OPTIMIZATION PROBLEMS

The optimization problems (8), (9), (12) and (13) are in general non-convex. We take optimization problem (9) as an example to present how to numerically solve these optimization problems. The variables of the optimization are  $\theta_A$ ,  $\theta_B$ ,  $x_t$ ,  $y_t$ ,  $t = 1, \dots, t_{\max}$ , degree distributions  $\Psi_A$  and  $\Psi_B$ . Consider the non-linear constraint

$$(\alpha_A + \alpha_{A+B} \Psi_B(y_{t-1})) \Psi'_A(x) + \theta_A \ln(1-x) \geq 0, \quad \forall x \in (x_{t-1}, x_t]. \quad (29)$$

Since it is impossible to check the inequality for all  $x \in (x_{t-1}, x_t]$ , we interpolate a number of  $M$  (e.g., 20) points that are evenly distributed in  $(x_{t-1}, x_t]$ , and force them to satisfy the above inequality. The same relaxation is applied to other non-linear constraints. We then solve this (relaxed) optimization using a non-linear optimization solver.<sup>4</sup>

Due to the relaxation, however, the outputs of the optimization solver may not all be feasible for the original optimization. For example, in (29), even when the  $M$  interpolated points satisfy the inequality, it is possible that there exist some other points in the line segment  $(x_{t-1}, x_t]$  violating the inequality. This tends to happen especially when  $x_t - x_{t-1}$  is large. Fig. 6 illustrates such an example. The region below the dotted curve and left of the solid curve is  $(\theta_A, \theta_B)$ -feasible. But there are two disjoint  $(\theta_A, \theta_B)$ -feasible regions. The first intersection of these two curves is not the target point  $(\eta_A, \eta_B)$ . Fig. 3 plots the curves for a feasible output of the optimization solver for the same values of  $\alpha_A$ ,  $\alpha_B$ ,  $\eta_A$  and  $\eta_B$ , where the degree distributions are

$$\begin{aligned} \Psi_A(x) &= 0.1040x + 0.8362x^2 + 0.0582x^{26} + 0.0007x^{27}, \\ \Psi_B(x) &= 0.1133x + 0.7902x^2 + 0.0662x^{13} + 0.0284x^{14} + 0.0020x^{15}. \end{aligned}$$

Therefore, we need to verify the feasibility of each output of the optimization solver. Though it is possible to increase the chance of obtaining feasible outputs by using larger values of  $M$  and  $t_{\max}$ , the optimization solver will run longer time. For example, we use  $M = 20$  and  $t_{\max} = 20$  for the results in Table I. Instead of using larger  $M$ , we add constraints to avoid a large jump from  $x_{t-1}$  to  $x_t$ . For those outputs that are not feasible, we may reduce the value of  $\theta_A$  and  $\theta_B$  a little bit to make the solution feasible.

Since those optimization problems are non-convex, we may not obtain the global optimal value. Therefore, we run the optimization solver multiple times (with randomly selected initial point) and pick the best among all the outputs. For the parameters we have evaluated, the feasible outputs returned by the optimization solver are all very close to the theoretical upper bound.

## REFERENCES

- [1] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. MobiCom '06*, New York, NY, USA, 2006.
- [2] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6463–6486, 2011.

<sup>4</sup>We use the *fmincon* function provided in Matlab with the active-set algorithm.

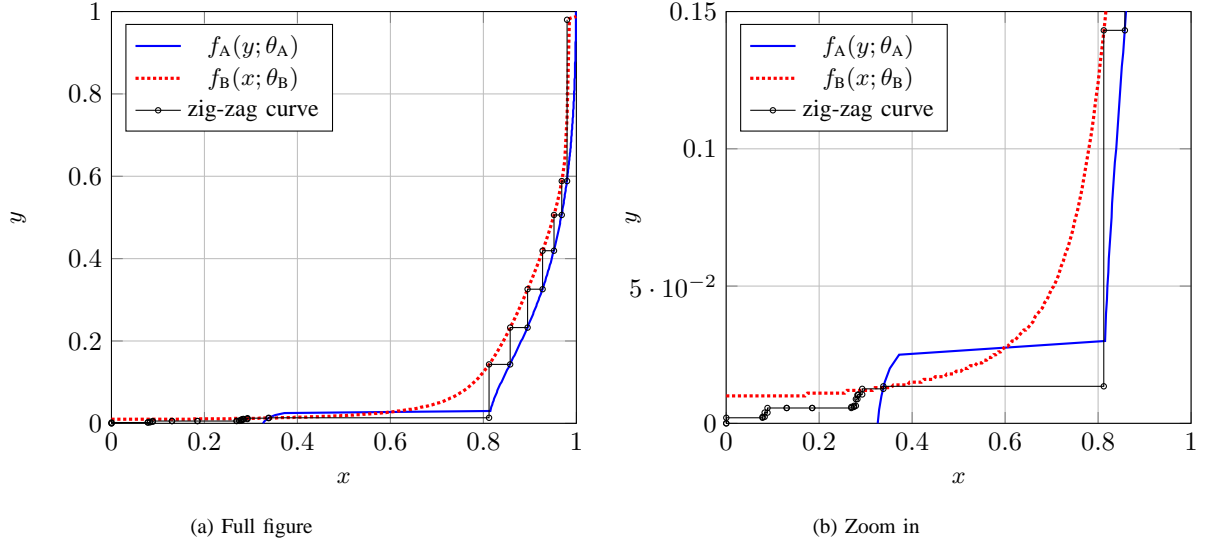


Fig. 6: Curves  $x = f_A(y; \theta_A)$  and  $y = f_B(x; \theta_B)$  with  $\alpha_A = \alpha_B = 0.25$  and  $\eta_A = \eta_B = 0.98$ , where  $\theta_A, \theta_B, x_t, y_t, t = 1, \dots, t_{\max}$ ,  $\Psi_A$  and  $\Psi_B$  are returned by the optimization solver. The zig-zag curve formed by line segments  $(x_t, y_t) - (x_{t+1}, y_t) - (x_{t+1}, y_{t+1})$ ,  $t = 0, 1, \dots, t_{\max} - 1$ . Here  $M = 10$ .

- [3] L. Lu, L. You, and S. C. Liew, "Network-coded multiple access," 2014, accepted by IEEE Trans. Mobile Computing, early access available at IEEE Xplore. [Online]. Available: <http://arxiv.org/abs/1307.1514>
- [4] L. You, S. C. Liew, and L. Lu, "Network-coded multiple access II: Toward realtime operation with improved performance," 2014. [Online]. Available: <http://home.ie.cuhk.edu.hk/~yl013/docs/NCMA2draft.pdf>
- [5] G. Cocco and S. Pfletschinger, "Seek and decode: Random multiple access with multiuser detection and physical-layer network coding," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014.
- [6] S. Verdú, *Multiuser detection*. Cambridge university press, 1998.
- [7] C. Feng, D. Silva, and F. Kschischang, "An algebraic approach to physical-layer network coding," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7576–7596, Nov 2013.
- [8] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva, "Communication over finite-ring matrix channels," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2890–2894.
- [9] M. Luby, "LT codes," in *Proc. 43rd Ann. IEEE Symp. on Foundations of Computer Science*, Nov. 2002, pp. 271–282.
- [10] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
- [11] S. Zhang and S.-C. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 5, pp. 788–796, June 2009.
- [12] D. Wubben and Y. Lang, "Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–5.
- [13] S. C. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: Tutorial, survey, and beyond," (*invited paper*) *Physical Communication*, vol. 6, pp. 4–42, 2013.
- [14] J. Zhu and M. Gastpar, "Gaussian (dirty) multiple access channels: A compute-and-forward perspective," in *Information Theory (ISIT), 2014 IEEE International Symposium on*, June 2014, pp. 2949–2953.
- [15] —, "Multiple access via compute-and-forward," *arXiv preprint arXiv:1407.8463*, 2014.
- [16] S. Puducheri, J. Klierer, and T. E. Fuja, "The design and performance of distributed LT codes," *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3740–3754, 2007.
- [17] B. Hern and K. Narayanan, "Joint compute and forward for the two way relay channel with spatially coupled LDPC codes," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012.

- [18] E. Casini, R. De Gaudenzi, and O. R. Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 4, pp. 1408–1419, 2007.
- [19] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *Communications, IEEE Transactions on*, vol. 59, no. 2, pp. 477–487, 2011.
- [20] E. Paolini, G. Liva, and M. Chiani, "High throughput random access via codes on graphs: Coded slotted ALOHA," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.
- [21] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA protocol for wireless networks," *Communications Letters, IEEE*, vol. 16, no. 12, pp. 2087–2090, 2012.
- [22] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted aloha: An optimal uncoordinated transmission policy," in *Turbo Codes and Iterative Information Processing (ISTC), 2012 7th International Symposium on*. IEEE, 2012, pp. 136–139.
- [23] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *arXiv preprint arXiv:1401.1626*, 2014.
- [24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc, 2006.
- [25] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of Random Processes via And-Or Tree Evaluation," in *SODA*, 1998, pp. 364–373.
- [26] A. Shokrollahi and M. Luby, *Raptor Codes*, ser. Foundations and Trends in Communications and Information Theory. now, 2011, vol. 6.